Wangkongqiang at HOPE-EXP@IberLEF 2026: Outcome-Oriented Expectation Analysis in Social Media

Kongqiang Wang[1,*], Peng Zhang[1] and Qingli Tan[2]

[1]School of Information Science and Engineering, Yunnan University, 650500, Kunming, Yunnan, China.
[2]College of Ecology and Environment, Yunnan University, 650500, Kunming, Yunnan, China.

Abstract

According to a recent report by the World Health Organization, there are thousands of people in the world suffering from a hopelessness feeling. The HOPE-EXP organization at IberLEF 2026 considers that hopeful expression identification is a key effective intervention to prevent these problems. These tasks we participated in were (1) Task A: Primary Label Classification (Single-label). This is a multi-class classification task aimed at determining whether a statement expressing a desire or expectation for a future outcome based on their posts, and then determining which kind of hope the content of the post belongs to. (2) Task B: Trigger Emotion Detection (Multi-label). This is a multi-label classification task aimed to identify one or more emotions associated with the post. (3) Task C: Span Extraction (Outcome Identification). The objective is to extract up to 3 spans that describe the expected or avoided outcome. We compare the performance of different machine learning (ML) model approaches and using two different vectorizers as processing input text to a classifier. Our final experimental best result is Overall 0.61921, F1_primary 0.8289, F1_emotion 0.65456, ROUGE1_span 0.37416. Our overall work results ranked 6th in the leaderboard list. The code of our project can be found at https://github.com/WangKongQiang/IberLEF_2026_HOPE-EXP.

Keywords

Hopeful Expression, Machine Learning, Multi-class Classification, Multi-label Classification

1. Introduction

Hope is not a simple phenomenon [1]. In real-world discourse, expressions of hope: May be explicit or implicit; Can include specific expected outcomes; May involve different actors (self, others, systems); Are often accompanied by complex emotional states. Thus, it is becoming increasingly important to evaluate the use of new technologies to assess the understanding of hopeful expressions in social media posts and the emotion needs of the population [2].

At the same time, social media platforms such as Telegram have become a popular way for people to express their feeling and emotions. Telegram is a free, end-to-end encrypted messaging service that allows users to send and receive messages and media files in private chats or groups that can be focused on particular topics and allow any user to observe or actively participate [3]. These characteristics make Telegram a suitable source for text-mining. There are also some other microblogging platforms. For instance, social media platforms such as Sina Weibo and WeChat Moments allow users to share posts, thereby also conveying daily news content. The official HOPE-EXP task description and dataset overview [4] indicate that the source data comes generically from multilingual social media. This is more shared and open compared to the Telegram platform.

With this context, an interesting approach is to use Natural Language Processing (NLP) techniques to analyze the language used by people who express expectations and emotional content [5]. Thus, we discover patterns that can be used to identify them and provide the necessary support. The HOPE-EXP [4] task at IberLEF 2026 [6] aims to promote the development of NLP solutions specifically for multi-language social media in terms of English (EN) and Spanish (ES) [7]. They propose four main

IberLEF 2026, September 2026, León, Spain
*Corresponding author.
wangkongqiang60@gmail.com (K. Wang); zpp1219@gmail.com (P. Zhang); tanqingli@stu.ynu.edu.cn (Q. Tan)
https://github.com/WangKongQiang/ (K. Wang); https://github.com/zpp1219/ (P. Zhang)
0009-0003-8736-6397 (K. Wang)

areas of focus for Single-label: Primary Label Classification (Task A), Multi-label: Trigger Emotion Detection (Task B), Outcome Identification: Span Extraction (Task C), Conditional: Outcome Role Classification (Task D). The official overall score is: Overall = Mean (TaskA_MacroF1, TaskB_MacroF1, TaskC_ROUGE1, TaskD_Stance_MacroF1, TaskD_Actor_MacroF1).

In this work, we present our proposed methods to Task A, B, C, and D of the 2026 edition of HOPE-EXP. This task involves Sentence-level classification; Multi-label emotion detection; Span extraction; Role-based semantic interpretation. These subtasks based on user posts within hope-expression-focused groups. Task A is split into six predictive labels according to the type of output required. Classify the post into:

- `Realistic Hope`: A statement expressing a desire or expectation for a future outcome that is plausible, achievable, or grounded in real-world conditions, even if uncertain.
- `Unrealistic Hope`: A statement expressing a desire or expectation for a future outcome that is highly improbable, impossible, fantastical, or disconnected from realistic constraints.
- `General Hope`: A vague or broad expression of optimism or positive expectation about the future without specifying a concrete, clearly defined outcome.
- `Hopelessness`: A statement indicating the absence of hope, characterized by pessimism, resignation, or the belief that no positive change or desirable outcome will occur.
- `Sarcastic Hope`: An ironic or mocking expression that superficially appears to convey hope or expectation but actually implies criticism, disbelief, or the opposite sentiment.
- `Not Hope`: A statement that does not express any desire, expectation, or anticipation of a future outcome, including neutral descriptions, factual statements, opinions, emotions, or narratives without forward-looking intent.

Task B is multi-label trigger emotion detection to identify one or more emotions associated with the post: sadness, joy, love, anger, fear, surprise, neutral/unclear. Rules based on multi-label allowed.

Task C is span extraction for outcome identification. If the primary label is any of hope categories, systems must extract up to 3 spans that describe the expected or avoided outcome. Each span must: Be an exact substring from the input text; Correspond to a desired or avoided outcome. If primary_label is *Not Hope* or *Hopelessness*, span_annotations must be an empty list. Span detection is evaluated using: *ROUGE 1*.

Task D is conditional to outcome role classification. For aligned predicted spans: outcome_stance: *Desired* vs *Avoided*; actor: *Self*, *Other*, *World/System*, *Unclear*. Only computed when spans are aligned.

Our main contributions and findings can be then summarized threefold:

- `Concatenate Posts for Text Processing.`
  The first thing we did was processing all the posts by the row_id. They used different languages (EN and ES) and concatenated their language expression content (include title and selftext) into a single text string, obtaining a total of 4857 text strings (one text strings per jsonl item) from train dataset. These text strings were done to obtain a single text representation of each user's post from which the labels were assigned. Thus, these text strings are able to be used as input text for the different machine learning (ML) models.
- `Machine Learning (ML) Models for Model Fitting.`
  We used different machine learning models, and they were LogisticRegression (*LR*), LinearSVC, RandomForestClassifier [8], Multinomial Naive Bayes (*MultinomialNB*), LinearSVC_TF-IDF_trick, GradientBoostingClassifier, ExtraTreesClassifier, KNeighborsClassifier (*KNN*), RidgeClassifier [9], Passive Aggressive Classifier, SGDClassifier, DecisionTreeClassifier, AdaBoostClassifier, BaggingClassifier, Bernoulli Naive Bayes (*BernoulliNB*), Complement Naive Bayes (*ComplementNB*), Gaussian Naive Bayes (*GaussianNB*), HistGradientBoostingClassifier, XGBClassifier, XGBClassifier_parameter, XGBClassifier_parameter_sparse_matrix, LGBMClassifier, CatBoost, NGBoost, TabNetClassifier, FastText, FastText_parameter, Vowpal Wabbit, Vowpal Wabbit_parameter, ThunderSVM, ThunderSVM_HashingVectorizer, H2O AutoML and AutoGluon respectively. The model we used in a development deployment version of Python 3.8

for detecting hope expression behavior or emotional states from text strings in Spanish (ES) or English (EN). We chose these models due to the outstanding performance previously for a text classification task [10] that shares characteristics similar to this one. Most references of the implementations of these mentioned machine learning (ML) models can be found in the Scikit-Learn [11] documentation.

- `Perform Text Vectorization if Classification Estimators is Required.`
  The text vector were obtained after concatenating the posts of each user into a single text string. The difference in performance between TF-idf Vectorizer, Hashing Vectorizer and the other vectorizers can be selected by the fact that we are taking advantage of the information gained before using machine learning (ML) model, which likely shares semantic similarities with our raw data content.

The rest of the paper is organized as follows: In the next section, we provide the relevant work and background for this task (**Section 2**). we analyze the dataset used for the specific task (**Section 3**). Then, we describe in detail our methodology for fitting and evaluating the models (**Section 4**). Finally, we discuss the results obtained (**Section 5**) and present our conclusions and future lines of work (**Section 6**).

2. Related Work

2.1. Hope and Future-Oriented Sentiment

Hope-related language understanding [12] lies at the intersection of sentiment analysis, emotion detection, and fine-grained pragmatic inference. Early work [13] in sentiment analysis primarily focused on polarity classification (*positive*, *negative*, *neutral*), but such formulations fail to capture nuanced future-oriented expressions such as hope, expectation, and pessimism. To address this limitation, recent studies [14] have explored richer affective representations, including multi-dimensional emotion taxonomies and cognitive appraisal frameworks [15].

Hope is a complex, future-directed cognitive-emotional state, often involving uncertainty and desirability [16]. Prior work has attempted to model hope as a distinct category within emotion classification, distinguishing it from related constructs such as optimism or desire. Computational approaches typically rely on supervised learning over annotated corpora, where hope expressions are identified based on linguistic cues such as modality ("*will*", "*might*"), desire verbs ("*wish*", "*hope*"), and conditional structures [17]. However, distinguishing between realistic and unrealistic hope remains underexplored, requiring models to incorporate commonsense reasoning and world knowledge to assess plausibility.

Recent advances [18] move beyond binary or ternary sentiment labels toward fine-grained taxonomies, similar to Task A in this work. Categories such as sarcastic hope and hopelessness are particularly challenging, as they involve implicit meaning, negation, or figurative language. Sarcasm detection has been widely studied, often leveraging contextual incongruity and pretrained language models [19], but integrating sarcasm with future-oriented intent classification is still an emerging area. Likewise, identifying general hope versus specific outcome-driven hope requires sensitivity to semantic specificity and event structure.

2.2. Emotion Detection and Multi-Label Learning

Task B aligns with the growing body of work on multi-label emotion classification, where a single text may express multiple co-occurring emotions. Benchmark datasets such as GoEmotions [20] have popularized fine-grained emotion annotation, and transformer-based architectures (e.g., BERT [21] and its variants [22]) have demonstrated strong performance in capturing contextual emotional cues. Multi-label settings introduce challenges such as label correlation and class imbalance, often addressed through threshold tuning, label-wise attention, or specialized loss functions.

Table 1
The official train and test dataset were provided.

Dataset Type	Language	Number of Posts	Percentage Proportion	Comments
Train Dataset	Overall	4857	100%	Posts used in fitting machine learning (ML) model.
	English (EN)	2243	46.18%	The content of the post is described in English.
	Spanish (ES)	2614	53.82%	The content of the post is described in Spanish.
Test Dataset	Overall	2082	100%	Posts used in testing machine learning (ML) model.
	English (EN)	995	47.79%	The content of the post is described in English.
	Spanish (ES)	1087	52.21%	The content of the post is described in Spanish.

2.3. Span Extraction and Rationalization

Task C relates to extractive rationale generation and span detection, where models identify text segments that justify predictions. This paradigm has gained traction for improving interpretability and explainability in NLP systems. Techniques range from sequence labeling (e.g., BIO tagging) to pointer networks and extractive question answering formulations. Evaluation using overlap-based metrics such as ROUGE-1 [23] is common, though aligning predicted spans with gold annotations remains challenging due to variability in valid expressions.

2.4. Outcome Role and Semantic Role Labeling

Task D extends span extraction by requiring structured interpretation of outcomes, including stance (desired vs. avoided) and actor (self, other, world). This is closely related to semantic role labeling (SRL) and event extraction, where systems identify participants and their roles in events. Recent approaches [24] leverage joint modeling to simultaneously predict spans and their attributes, often using multi-task learning or structured prediction frameworks. Incorporating commonsense knowledge and discourse context is crucial for accurately identifying implicit actors or ambiguous roles.

Across all tasks, pretrained transformer models have become the dominant paradigm due to their ability to encode rich contextual semantics. Recent work [25] explores joint or multi-task learning setups that unify classification, emotion detection, and span extraction into a single framework, improving consistency across subtasks. Additionally, instruction tuning and prompt-based methods have shown promise in handling complex, multi-component annotations with limited supervision. In summary, while substantial progress has been made in sentiment and emotion analysis [26], the integrated modeling of fine-grained hope categories, multi-label emotions, outcome spans, and their semantic roles remains relatively underexplored. This work contributes to bridging these areas by formulating a comprehensive, multi-task machine learning (ML) framework for hope understanding.

3. Dataset

The dataset given for the HOPE-EXP task consisted of a total of thousands of individual post from two different user languages (see Table 1), each language with a variable number of posts. The annotation process consisted of labeling each user based on the evidence from their post content of hopeful expressions in social media [27]. Thus, a total of six labels were used for the task A. Each post was asked to assign one of the following six labels: *Realistic Hope*, *Unrealistic Hope*, *General Hope*, *Hopelessness*, *Sarcastic Hope*, and *Not Hope*.

To increase the accuracy of machine learning (ML) model prediction, we attempt to fitting model accompanied to multi-label emotional states detection at the same time. We used two different forms of text vectorization representation rather than the original data text as model input (see Table 2).

4. Methodology

We evaluate different machine learning (ML) techniques to solve these subtasks. Our main predictive-modeling approaches were explored in the field of machine learning (ML), combining the automatic

Table 2
Two different scenarios of hyperparameter for text vectorization representation.

Text Representation	Parameter	Value	Comments
TfidfVectorizer	max_features	2000	Limit the number of features
	min_df	2	It appears at least twice
	max_df	0.9	Filter high-frequency words
	ngram_range	(1,1)	Only use unigram
HashingVectorizer	n_features	4096	Characteristic dimension
	alternate_sign	False	Avoid symbolic conflicts
	ngram_range	(1,1)	Only use unigram
	norm	l2	Normalization

Table 3
The number of labels in the training dataset used in our experiments for the two subtasks, as well as the corresponding percentage ratios.

Primary_label Distribution	Trigger_emotions Distribution
Not Hope: 1400 (28.82%)	sadness: 2438 (31.32%)
General Hope: 937 (19.29%)	anger: 1335 (17.15%)
Realistic Hope: 701 (14.43%)	neutral/unclear: 1277 (16.41%)
Sarcastic Hope: 700 (14.41%)	fear: 1119 (14.38%)
Unrealistic Hope: 700 (14.41%)	joy: 614 (7.89%)
Hopelessness: 419 (8.63%)	love: 611 (7.85%)
	surprise: 389 (5.00%)
Total: 4857 (100%)	Total: 7783 (100%)

training of multiple machine learning (ML) models and development frameworks with automatic parameter tuning mechanisms from the user's posts as text features. The following section describes the steps taken for each machine learning (ML) approach, first describing how the data was preprocessed and later explaining the fitting and evaluation process done for these subtasks.

4.1. Data Processing

Based on the detailed description and practical examples provided in the previous part of the article, a brief account will be given here, include:

- `Concatenated Posts` : According to the posts of the English (EN) or Spanish (ES) users, concatenate the title and selftext into a single text string and name it as the text column.
- `Text Vectorization Representation` : Perform text vectorization representation on the posts obtained by concatenating title columns and selftext columns from the train dataset and the test dataset [1].

To prepare for fitting different machine learning (ML) models, the original training data was not split into train and validation datasets, all the training data are used for model fitting. This is done to enable the machine learning (ML) model to learn more data vectorization representations.

4.2. Solving Substask B: Multi-label Trigger Emotion Detection by Using OneVsRestClassifier (OVR)

By the label distribution discussion of task A and task B in Table 3, it should be clear to see that all labels of subtask B: trigger emotion detection give an imbalance situation in the available train dataset. This observation led us to consider using machine learning (ML) models that solve for this one subtask by using OneVsRestClassifier.

OneVsRestClassifier (abbreviated as OvR or One-vs-All) is the most commonly used strategy in multi-classification problems, and its core idea is very intuitive: Break down a multi-classification problem into multiple binary classification problems for solution.

[1]We can enable the display and use of raw data content only the official dataset is provided and no additional extended datasets were used.

Fundamental. Suppose you have A K-class classification problem (for example, three Classes: A/B/C), OneVsRestClassifier will: Train K binary classification models; Each model is responsible for one class vs. all other classes. The detailed explanation of the specific situation is shown in Table 4.

Table 4
One-vs-Rest strategy for a three classes problem

Classifier	Positive Class (1)	Negative Class (0)
Model 1	A	B, C
Model 2	B	A, C
Model 3	C	A, B

Forecasting Process. When you input a new sample: All K models will output a probability or score; Select the category with the highest score as the final prediction. In essence,

$$\hat{y} = \arg \max_{i \in \{1,\ldots,K\}} P(y = i \mid x) \tag{1}$$

Usage in Sklearn. OneVsRestClassifier is a wrapper that can be applied to any binary classification model.

Applicable Scene. Mainly applied in multi-class problems and multi-label classification.

- Multi-class Problems. Originally, it was a binary classification model (such as *LogisticRegression*, *SVM*). Task need to expand to multiple categories.
- Multi-label Classification. For example, a single text can simultaneously belong to multiple emotions (e.g. *fear* + *sadness* + *anger*).

OvR is the default multi-label method supported in sklearn.

Advantage. Simple and intuitive; It can be extended to any binary classification model; Support multi-label tasks; The training speed is relatively fast (more economical than One-vs-One).

Disadvantage. The problem of category imbalance is obvious (1 vs rest); There is no modeling of the relationship between categories among different classifiers; When there are many categories, conflicts may occur (multiple classifiers all consider it a positive class).

Summary. OneVsRestClassifier Used K binary classification models to piece together a multi-classification or multi-label model.

4.3. Training a ML Classifier with Text Vectorization Representation

A text vectorization number list [28] is a semantically meaningful real-valued vectorization representation of a sentence, obtained from the outputs of a vectorizer model. The properties of this representation are so that sentences that express similar meanings are mapped closer to each other in the vectorization representation space. In this way, the process of encoding text as numeric vectors can be used directly to extract features for a classifier or regressor, which will try to learn from the semantic information of these encodings to predict the label of their corresponding posts. However, it note that this approach requires the need to have a vectorizer model to perform this encoding. Furthermore, it assumes that the vectorizer model will be good enough at capturing the semantic information of the posts given as model input, enough for the classifier or regressor to learn from it. Assuming that this is the case, this approach has the advantage that it is much faster to train these kinds of classifiers or regressors with regular CPUs, with the most time-consuming part being obtaining the embeddings of the training or evaluation posts, which only has to be done once. However, it is necessary to evaluate different vectorization encoding models and different prediction models (classifiers/regressors) to find the best combination for the task at hand.

As such, we conducted experiments using different vectorizer models to find the best encoding model. Particularly, we tested two different versions of vectorizers trained with different parameters

in train dataset. These vectorizer model parameters are described in Table 2. Additionally, we experimented with multiple different classifiers, including LogisticRegression (*LR*), LinearSVC, RandomForestClassifier, Multinomial Naive Bayes (*MultinomialNB*), LinearSVC_TF-IDF_trick, GradientBoostingClassifier [29], and so on. These machine learning (ML) models were chosen due to their ease of implementation and the fact that they are commonly used in the literature.

The process of training and evaluating these machine learning (ML) models as follows: First, the training set was encoded using the vectorizer models and the resulting embeddings were used as features for a ML classifier. The classifier was then fitted using the labels of Subtask A/B (Single-label/Multi-label Classification) and the resulting model was used to predict the labels of the test set. The predictions were then evaluated with the F1_primary/F1_emotion. This process was repeated for each combination of different vectorizer models and ML classifiers.

4.4. Fitting the Machine Learning Model for Classifier

Next, we will introduce the machine learning (ML) classifiers that we used in this classification task one by one.

LogisticRegression (LR). That is official baseline.

LinearSVC. LinearSVC is usually stronger than LR in text classification task.

Random Forest. It is a non-linear model. If you want to try a tree model. However, it should be noted that Random Forest usually performs worse than SVM/LR when dealing with high-dimensional TF-IDF data.

Multinomial Naive Bayes. It is a classic model for text processing, and this approach is usually a classic baseline for text tasks. Its advantages are: Fast speed; Friendly to TF-IDF; Often used as the baseline.

LinearSVC_TF-IDF_trick. Added specified parameters to the TfidfVectorizer. That is, we create TfidfVectorizer used parameters of *ngram_range = (1,2), min_df = 2, max_df = 0.9*.

Gradient Boosting. Gradient boosting trees are suitable for small to medium-sized datasets. However, it should be noted that it generally performs poorly in the case of high-dimensional sparse TF-IDF.

Extra Trees. Extreme Random Forest is usually more random than RandomForest. Its features are: Training speed is fast; Robust to noise.

K-Nearest Neighbors (KNN). Simple but sometimes surprisingly useful. It disadvantages are: High-dimensional TF-IDF is slow in speed; High memory usage.

Ridge Classifier. It is a linear model. It is actually a L2 regularized linear classifier, and its performance is often comparable to that of *SVM*. Its advantages are: Fast speed; It is very good for the text data.

Passive Aggressive Classifier. This is a linear model specifically designed for text classification. Its features are: Very suitable for sparse TF-IDF; Training speed is fast.

SGDClassifier. It is a linear model, and many NLP baselines use this one. Its advantages are: Support large-scale data; The effect is similar to *SVM*.

Decision Tree. Characteristics of *DecisionTreeClassifier*: Highly interpretable. However, its effect on high-dimensional text features is generally not good.

AdaBoost. *AdaBoost* is one of the Boosting methods. Characteristics of *AdaBoostClassifier*: Capable of handling nonlinearity; Sensitive to noise.

BaggingClassifier. It employs the Bagging ensemble method. Characteristics of *BaggingClassifier*: Reduce overfitting; Good stability.

Bernoulli Naive Bayes. It would be great if TF-IDF were converted to binary features. *BernoulliNB* features are: Has a good effect on short texts; At an extremely fast speed.

Complement Naive Bayes It is commonly used in text tasks and is a modified Naive Bayes specifically designed for text classification. Its advantages are: More stable in dealing with class imbalance; Common in NLP papers.

Gaussian Naive Bayes. However, note that *GaussianNB* does not support sparse matrix. It is generally not recommended for TF-IDF.

HistGradientBoosting. It is the new generation of GBDT. Its features are: Faster than traditional GBDT; But it is still not satisfactory for sparse text.

The above classification method mainly relies on the Sklearn dependency package. In the following part, we will attempt more machine learning (ML) models beyond Sklearn dependency package.

XGBoost. Its advantages are: Technically mature; Commonly used in *Kaggle*; It works very well on structured features.

XGBClassifier_parameter. A more stable way of writing *XGBoost* is to explicitly specify the parameters. We explicitly specify the parameters for *XGBClassifier* as *objective = "multi:softmax", num_class = len(le.classes_), eval_metric = "mlogloss"*.

XGBClassifier_parameter_sparse matrix. Previously, the input for using *XGBoost* was TF-IDF-Dense-XGBoost. Usually, it is not recommended to convert to dense format, as it consumes a lot of memory. *XGBoost* can directly process sparse matrices.

LightGBM. Its advantages are: Training speed is fast; Low memory usage; Friendly to high-dimensional features.

CatBoost. It is friendly to text processing. Its advantages are: Very good for categorical features; The default parameters are very powerful.

NGBoost. Its features are: Probabilistic gradient boosting; Can output probability distribution.

TabNet. Advantages of deep learning table (*TabNet*) model: Utilize attention; Suitable for structured data. We employed a deep learning table (*TabNet*) model and *TruncatedSVD* for dimensionality reduction. The provided parameters were *n_components = 300*.

FastText. Facebook Text Classification. Its advantages are: Specialized in text classification; Very fast. Its features are: Directly uses the text, no need for TF-IDF.

fasttext_parameter. *FastText* employs recommended parameters, thereby enhancing its performance. In the *train_supervised* function of fasttext, use the parameter as *input = FASTTEXT_TRAIN. epoch = 50, lr = 0.3, wordNgrams = 3, dim = 200, loss = "softmax"*.

Vowpal Wabbit. Its advantages are: Ultra-large-scale data as predictive knowledge; Online learning.

Vowpal Wabbit_parameter. Use the enhanced parameters of Vowpal Wabbit (*VW*), and specify the parameters within the *Workspace* function *- -oaa {num_class} - -loss_function logistic - -ngram 2 - -skips 1 -b 26 -l 0.5 - -adaptive - -invariant - -quiet*.

ThunderSVM. Its features are: GPU-accelerated SVM; Sklearn API compatibility. Text vectorization is performed using *TfidfVectorizer*, and dimensionality reduction is achieved through *TruncatedSVD*. Its parameter is *n_components = 300*. The parameters of the used *SVC* are *kernel='rbf'*.

ThunderSVM_HashingVectorizer. Text vectorization is performed using *HashingVectorizer*, and dimensionality reduction is achieved through *TruncatedSVD*. Its parameter is *n_components = 300*. The parameters of the used *SVC* are *kernel='linear'*.

Apart from the approach mentioned above, we also experimented with the *H2O AutoML* or *AutoGluon* approach of automatic parameter tuning mechanisms and automatic training of multiple machine learning (ML) models with the labels of the corresponding subtasks. The machine learning (ML) models were fitted using an 12th Gen Intel(R) Core(TM) i7-12700KF CPU and NVIDIA GeForce RTX 3090 24G GPU, where the parameters of the *AutoGluon* machine learning (ML) model, see Table 5.

5. Results

Using the ML approaches mentioned in the prior section, we came up with different machine learning (ML) models to solve the subtask A and subtask B of HOPE-EXP@IberLEF 2026: Outcome-Oriented Expectation Analysis in Social Media. The results in this section are obtained from selecting the best-performing models after evaluating the different ML approaches and hyperparameters on the train dataset. The final predictions were obtained from a test dataset of posts from 2082 jsonl items never observed during the training process and evaluated against the task's true labels.

Table 5
Hyperparameters of the machine learning (ML) model for classification tasks used in AutoGluon.

ML Model	Hyperparameter	Value
NN_TORCH	activation	elu
	dropout_prob	0.10078
	hidden_size	108
	learning rate	0.00274
GBM	extra_trees	True
	ag_args	{'name_suffix': 'XT'}
CAT	depth	6
	grow_policy	SymmetricTree
	l2_leaf_reg	2.15428
	learning_rate	0.06864
XGB	colsample_bytree	0.69173
	enable_categorical	False
	learning_rate	0.01806
	max_depth	10
FASTAI	bs	256
	emb_drop	0.54118
	epochs	43
	layers	[800, 400]
	lr	0.01520
	ps	0.23783
RF	criterion	gini
	ag_args	{'name_suffix': 'Gini', 'problem_types': ['binary', 'multiclass']}
XT	criterion	gini
	ag_args	{'name_suffix': 'Gini', 'problem_types': ['binary', 'multiclass']}
KNN	weights	uniform
	ag_args	{'name_suffix': 'Unif'}

In the Table 6 below, we report the relevant metrics obtained for these subtask A, B, C and D compare them against the ones obtained from baseline models provided by the organizers of the competition. In particular, we report both absolute classification task metrics, obtained after observing all the posts of each jsonl item, and span detection is evaluated using *ROUGE 1* metrics, obtained after incrementally observing the posts extract up to 3 spans that describe the expected or avoided outcome. Additionally, Table 6 displays the overall average score of the four subtasks. For Task C, we reused the baseline output provided by the organizers rather than developing a custom method. We did not attempt Task D because of time and resource constraints, and instead focused on improving the performance of the tasks we participated in.

6. Conclusions

The results show that the different machine learning (ML) approaches considered in this work were successful at modeling of the predictive subtasks, with at least one of our models outperforming the baselines in most cases. Thus, it may be important to explore different ML fitting approaches to improve the performance of hopeful expression classification or emotional states detection. This might include directly employing online learning ML model to predict and update the model as new posts come in or incorporating an ensemble of models to make independent decisions about text content of a post and combining them for a final decision. Additionally, we may also look into more efficient implementations of the transformer approach to minimize the errors compared to machine learning (ML) models. These improvements are crucial when considering the deployment of our models in real-world situations and will be the focus of future work.

7. Acknowledgments

Thank you for Hope-Exp@IberLEF 2026 organizing the competition and providing the official train and test dataset and other support, and thanks to students of Yunnan University individuals and groups that assisted in the research and the preparation of the system environment work.

Table 6
Official experimental results from various machine learning (ML) models on a test set. The bolded experimental results represent the official baseline experiment.

ML model	Overall	F1_primary	F1_emotion	ROUGE1_span	F1_stance	F1_actor
HistGradientBoosting	0.61921	0.8289	0.65456	0.37416	n/a	n/a
LightGBM	0.61844	0.82609	0.65506	0.37416	n/a	n/a
XGBoost	0.61009	0.8094	0.64672	0.37416	n/a	n/a
XGBClassifier_parameter	0.61009	0.8094	0.64672	0.37416	n/a	n/a
CatBoost	0.60972	0.80982	0.64517	0.37416	n/a	n/a
XGBClassifier_parameter_sparse_matrix	0.60569	0.80521	0.6377	0.37416	n/a	n/a
Passive Aggressive Classifier	0.60552	0.83045	0.61195	0.37416	n/a	n/a
SGDClassifier	0.60456	0.84202	0.59751	0.37416	n/a	n/a
LinearSVC	0.60079	0.84941	0.5788	0.37416	n/a	n/a
H2O AutoML	0.59901	0.81323	0.60963	0.37416	n/a	n/a
LinearSVC_TF-IDF_trick	0.59145	0.85793	0.54226	0.37416	n/a	n/a
GradientBoosting	0.58767	0.76322	0.62562	0.37416	n/a	n/a
Vowpal Wabbit_parameter	0.58359	0.82594	0.55066	0.37416	n/a	n/a
Ridge Classifier	0.5807	0.84415	0.52378	0.37416	n/a	n/a
Bagging	0.56012	0.6687	0.63752	0.37416	n/a	n/a
Vowpal Wabbit	0.55614	0.79029	0.50398	0.37416	n/a	n/a
AutoGluon	0.54451	0.81725	0.44211	0.37416	n/a	n/a
AdaBoost	0.51963	0.55268	0.63207	0.37416	n/a	n/a
LogisticRegression (LR)(baseline experiment)	**0.51663**	**0.82514**	**0.3506**	**0.37416**	**n/a**	**n/a**
Decision Tree	0.50567	0.5624	0.58044	0.37416	n/a	n/a
NGBoost	0.50277	0.64303	0.49111	0.37416	n/a	n/a
RandomForest	0.48451	0.72937	0.35001	0.37416	n/a	n/a
ThunderSVM_HashingVectorizer	0.48008	0.78014	0.28593	0.37416	n/a	n/a
Extra Trees	0.47929	0.74152	0.32218	0.37416	n/a	n/a
FastText	0.45841	0.82892	0.0578	0.379	1.0	0.02632
FastText_parameter	0.45649	0.81935	0.0578	0.379	1.0	0.02632
BernoulliNB	0.42304	0.55469	0.34026	0.37416	n/a	n/a
ComplementNB	0.40758	0.64799	0.20059	0.37416	n/a	n/a
K-Nearest Neighbors (KNN)	0.40242	0.49521	0.33791	0.37416	n/a	n/a
GaussianNB	0.37556	0.49202	0.2605	0.37416	n/a	n/a
MultinomialNB	0.32626	0.4144	0.19023	0.37416	n/a	n/a
ThunderSVM	0.2499	0.07457	0.30098	0.37416	n/a	n/a
TabNet	0.24263	0.19089	0.16282	0.37416	n/a	n/a

8. Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] F. Balouchzahi, G. Sidorov, A. Gelbukh, Polyhope: Two-level hope speech detection from tweets, Expert Syst. Appl. 225 (2023). URL: https://doi.org/10.1016/j.eswa.2023.120078. doi:10.1016/j.eswa.2023.120078.

[2] A. M. Mármol-Romero, P. Álvarez-Ojeda, A. Moreno-Muñoz, F. M. P. del Arco, M. D. Molina-González, M.-T. Martín-Valdivia, L. A. Ureña-López, A. Montejo-Ráez, Overview of mentalriskes at iberlef 2025: Early detection of mental disorders risk in spanish, Procesamiento del Lenguaje Natural 75 (2025).

[3] Álvarez-Ojeda, Pablo, Cantero-Romero, M. Victoria, Semikozova, Anastasia, Montejo-Ráez, Arturo, The precom-sm corpus: Gambling in spanish social media, in: Proceedings of the 31st International Conference on Computational Linguistics, 2025, pp. 17–28.

[4] F. Balouchzahi, S. Butt, H. Gómez-Adorno, S. M. Jiménez-Zafra, H. G. Ceballos, G. Sidorov, Overview of hope-exp at iberlef 2026: Outcome-oriented expectation analysis in social media, Procesamiento del Lenguaje Natural 77 (2026).

[5] S. Butt, F. Balouchzahi, M. Amjad, S. M. Jiménez-Zafra, H. G. Ceballos, G. Sidorov, Overview of polyhope at iberlef 2025: Optimism, expectation or sarcasm?, Procesamiento del lenguaje natural 75 (2025) 461–474.

[6] A. Bonet-Jover, J. Á. González-Barba, L. Chiruzzo, Overview of IberLEF 2026: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian

Languages Evaluation Forum (IberLEF 2026), co-located with the 42nd Conference of the Spanish Society for Natural Language Processing (SEPLN 2026), CEUR-WS.org, 2026.

[7] S. Butt, F. Balouchzahi, A. Amjad, M. Amjad, H. Ceballos, S. M. Zafra, Optimism, expectation, or sarcasm? multi-class hope speech detection in spanish and english (2025). doi:10.13140/RG.2.2.19761.90724.

[8] L. Breiman, Random forests, in: Machine Learning, volume 45, 2001, p. 5–32. URL: https://doi.org/10.1023/A:1010933404324. doi:10.1023/A:1010933404324.

[9] A. E. Hoerl, R. W. Kennard, Ridge regression: Biased estimation for nonorthogonal problems, in: Technometrics, 55–67, [Taylor & Francis, Ltd., American Statistical Association,American Society for Quality], 1970, p. 12. URL: https://www.jstor.org/stable/1267351. doi:10.2307/1267351.

[10] González-Barba, J. Ángel, Chiruzzo, Luis, Jiménez-Zafra, S. María, Overview of IberLEF 2025: Natural Language Processing Challenges for Spanish and other Iberian Languages, in: Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2025), co-located with the 41st Conference of the Spanish Society for Natural Language Processing (SEPLN 2025), CEUR-WS. org, 2025.

[11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R.Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in python, in: Journal of Machine Learning Research, 12, 2011, p. 2825–2830.

[12] G. Sidorov, F. Balouchzahi, L. Ramos, H. Gómez-Adorno, A. Gelbukh, Multilingual identification of nuanced dimensions of hope speech in social media texts, Scientific Reports 15 (2025) 26783.

[13] J. Xiong, O. Lipsitz, F. Nasri, L. M. W. Lui, H. Gill, L. Phan, D. Chen-Li, M. Iacobucci, R. Ho, A. Majeed, R. S. McIntyre, Impact of covid-19 pandemic on mental health in the general population: A systematic review, in: Journal of Affective Disorders 277, 2020, p. 55–64. URL: https://www.sciencedirect.com/science/article/pii/S0165032720325891. doi:10.1016/j.jad.2020.08.001.

[14] S. M. Jiménez-Zafra, M. Á. Garcia-Cumbreras, D. García-Baena, J. A. Garcia-Díaz, B. R. Chakravarthi, R. Valencia-García, L. A. Ureña-López, Overview of hope at iberlef 2023: Multilingual hope speech detection, Procesamiento del lenguaje natural 71 (2023) 371–381.

[15] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, J. B. L. Fang, S. Chintala, Pytorch: An imperative style, high-performance deep learning library, in: Advances in Neural Information Processing Systems 32, Curran Associates, Inc., 2019, p. 8024–8035. URL: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf.

[16] D. L. Padial, D. Gómez, hackathon-somos-nlp-2023-roberta-base-bne-finetuned-suicide-es· hugging face (2023). URL: https://huggingface.co/hackathon-somos-nlp-2023/roberta-base-bne-finetuned-suicide-es.

[17] C. C. Liu, J. Pfeiffer, I. Vulić, I. Gurevych, Improving generalization of adapter-based crosslingual transfer with scheduled unfreezing, 2023. URL: http://arxiv.org/abs/2301.05487.

[18] D. García-Baena, M. Á. García-Cumbreras, S. M. Jiménez-Zafra, J. A. García-Díaz, R. Valencia-García, Hope speech detection in spanish: The lgbt case, Language Resources and Evaluation 57 (2023) 1487–1514.

[19] A.G.Fandiño, J.A.Estapé, M. Pàmies, J.L.Palao, J.S.Ocampo, C.P.Carrino, C.A.Oller, C.R.Penagos, A.G.Agirre, M. Villegas, Maria: Spanish language models, Procesamiento del Lenguaje Natural 68 (2022). URL: https://upcommons.upc.edu/handle/2117/367156#.YyMTB4X9A-0.mendeley. doi:10.26342/2022-68-3.

[20] D. Demszky, D. Movshovitz-Attias, J. Ko, A. Cowen, G. Nemade, S. Ravi, GoEmotions: A dataset of fine-grained emotions, in: D. Jurafsky, J. Chai, N. Schluter, J. Tetreault (Eds.), Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, Online, 2020, pp. 4040–4054. URL: https://aclanthology.org/2020.acl-main.372/. doi:10.18653/v1/2020.acl-main.372.

[21] J. Devlin, M.-W. Chang, K. Lee, K. Toutanova, BERT: Pre-training of deep bidirectional transformers for language understanding, in: J. Burstein, C. Doran, T. Solorio (Eds.), Proceed-

ings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), Association for Computational Linguistics, Minneapolis, Minnesota, 2019, pp. 4171–4186. URL: https://aclanthology.org/N19-1423/. doi:10.18653/v1/N19-1423.

[22] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, V. Stoyanov, Roberta: A robustly optimized bert pretraining approach, 2019. URL: http://arxiv.org/abs/1907.11692. doi:10.48550/arXiv.1907.11692.

[23] C.-Y. Lin, ROUGE: A package for automatic evaluation of summaries, in: Text Summarization Branches Out, Association for Computational Linguistics, Barcelona, Spain, 2004, pp. 74–81. URL: https://aclanthology.org/W04-1013/.

[24] D. García-Baena, F. Balouchzahi, S. Butt, M. Á. García-Cumbreras, A. L. Tonja, J. A. García-Díaz, S. Bozkurt, B. R. Chakravarthi, H. G. Ceballos, R. Valencia-García, et al., Overview of hope at iberlef 2024: Approaching hope speech detection in social media from two perspectives, for equality, diversity and inclusion and as expectations, Procesamiento del lenguaje natural 73 (2024) 407–419.

[25] G. Sidorov, F. Balouchzahi, S. Butt, A. Gelbukh, Regret and hope on transformers: An analysis of transformers on regret and hope speech detection datasets, Applied Sciences 13 (2023) 3983.

[26] W. Zhang, M. Xu, Q. Jiang, Opinion mining and sentiment analysis in social media: Challenges and applications, in: F. F.-H. Nah, B. S. Xiao (Eds.), HCI in Business, Government, and Organizations, Springer International Publishing, Cham, 2018, pp. 536–548.

[27] F. Balouchzahi, S. Butt, M. Amjad, G. Sidorov, A. Gelbukh, Urduhope: Analysis of hope and hopelessness in urdu texts, Knowledge-Based Systems 308 (2025) 112746.

[28] C. S. Perone, R. Silveira, T. S. Paula, Evaluation of sentence embeddings in downstream and linguistic probing tasks, 2018. URL: http://arxiv.org/abs/1806.06259.

[29] J. Friedman, Greedy function approximation: A gradient boosting machine, in: The Annals of Statistics, 2000, p. 29. doi:10.1214/aos/1013203451.