Outcome-Oriented Hope Analysis in Social Media for HOPE-EXP 2026

Luis David Aguilar Colorado[1,*], Helena Montserrat Gómez Adorno[2]

[1]*Posgrado en Ciencia e Ingeniería de la Computación, Universidad Nacional Autónoma de México (UNAM), Ciudad de México, México*

[2]*Instituto de Investigaciones en Matemáticas Aplicadas y en Sistemas, Universidad Nacional Autónoma de México (UNAM), Ciudad de México, México*

Abstract

This paper describes the system developed for the HOPE-EXP 2026 shared task, which focuses on the analysis of hope expressions in social media posts. The task requires a structured output with four components: primary label classification, multi-label emotion detection, span extraction for expected or avoided outcomes, and semantic role classification through `outcome_stance` and `actor`.

The proposed system follows a modular architecture based on multilingual transformer models. For primary label classification, we fine-tuned `xlm-roberta-large`. Emotion detection was addressed as a multi-label classification problem using a weighted loss and decision thresholds calibrated by emotion and language. Span extraction was formulated as a sequence labeling task using a BIO scheme, followed by post-processing to ensure that the extracted spans were valid substrings of the input text. Finally, role classification was addressed with separate span-aware models for `outcome_stance` and `actor`, evaluated through a lightweight out-of-fold strategy.

The final submission achieved an official overall score of **0.78273** and ranked **2nd** on the HOPE-EXP leaderboard. The system obtained strong results in primary label classification, emotion detection, span extraction, and stance classification. Actor classification remained the most challenging component, partly because the actor may depend on contextual information that is reduced once a span has been extracted.

Keywords

Hope Speech Detection, Social Media Analysis, Multilingual NLP, Text Classification, Information Extraction, Transformers

1. Introduction

Automatic hope detection in social media is not limited to identifying positive language. A post may express an explicit desire, an implicit expectation, a situation the author wants to avoid, a sarcastic tone, or even a lack of hope. In addition, hope can be associated with different emotions and with different actors: the person who writes the post, other people, or broader social systems [1, 2].

The HOPE-EXP 2026 task proposes an outcome-oriented text analysis scenario, where the goal is not only to decide whether a post contains hope, but also to produce a structured interpretation of its content. The task is part of IberLEF 2026 and focuses on social media posts in a bilingual Spanish–English setting [1, 3]. The problem combines several tasks commonly addressed in natural language processing: text classification, multi-label classification, information extraction, sequence labeling, and context-dependent semantic classification.

Previous work on hope-related language has addressed several connected problems, including hope speech detection, expectation classification, optimism, sarcasm, hopelessness, and multilingual analysis in social media [4, 5, 6, 2, 7, 8, 9]. HOPE-EXP builds on this line of research, but extends it by requiring a structured output. Instead of predicting only a global label, systems must also identify emotions, extract textual outcomes, and assign semantic roles to the extracted spans.

The system developed in this work adopts a modular strategy. Instead of training a single model for all outputs, specialized components were designed for each subtask. This decision was motivated

IberLEF 2026, September 2026, León, Spain

*Corresponding author.

david.aguilar@unam.edu (L. D. A. Colorado); helena.gomez@iimas.unam.mx (H. M. G. Adorno)

by the different nature of each part of the challenge: Task A is a multi-class classification problem at the post level; Task B is a multi-label emotion classification problem; Task C requires locating exact spans within the text; and Task D classifies the stance and actor associated with each span. This decomposition allows each component to use the representation and post-processing strategy that best matches its output.

The rest of the paper is organized as follows. Section 2 describes the shared task. Section 3 presents the corpus. Section 4 describes the proposed system. Section 5 presents the experimental setup. Section 6 reports the internal results. Finally, Sections 7 and 8 discuss the main errors, limitations, and conclusions.

2. Task Description

Each HOPE-EXP instance contains an identifier, the language, the title, and the body of a post:

```
{
  "row_id": int,
  "lang": string,
  "title": string,
  "selftext": string
}
```

The system must generate a structured output with the primary label, the associated emotions, and, when applicable, a list of spans with semantic information:

```
{
  "row_id": int,
  "primary_label": "...",
  "trigger_emotions": [...],
  "span_annotations": [
    {
      "span": "...",
      "outcome_stance": "...",
      "actor": "..."
    }
  ]
}
```

The task is divided into four subtasks:

- **Task A: Primary Label Classification.** Multi-class classification of each post into one of six labels: *General Hope*, *Realistic Hope*, *Unrealistic Hope*, *Sarcastic Hope*, *Hopelessness*, and *Not Hope*.
- **Task B: Trigger Emotion Detection.** Multi-label emotion classification. The official emotions are `sadness`, `joy`, `love`, `anger`, `fear`, `surprise`, and `Nuetral/unclear`. The spelling `Nuetral/unclear` is kept because it is the label form used in the shared task files.
- **Task C: Span Extraction.** Extraction of up to three spans that describe the expected or avoided outcome. Each span must be an exact substring of the input text.
- **Task D: Outcome Role Classification.** Conditional classification over each extracted span. The system predicts `outcome_stance`, with values `Desired` or `Avoided`, and `actor`, with values `Self`, `Other`, `World/System`, or `Unclear`.

The official overall score is computed as the average of five components: Task A Macro-F1, Task B Macro-F1, Task C ROUGE-1, `outcome_stance` Macro-F1, and `actor` Macro-F1. Therefore, the system must maintain balanced performance across classification, extraction, and semantic analysis.

3. Dataset Description

The training set contains 4,857 examples, and the unlabeled test set contains 2,082. Each instance includes a post identifier, a language code, a title, and a field named `selftext`. In Reddit-style data,

`selftext` refers to the main body of the post, while `title` contains the headline or short introductory text. Some examples had an empty `selftext` field; these cases were kept because the title still provided useful information for classification.

To preserve as much context as possible, a new field called `full_text` was created by concatenating `title` and `selftext`. This representation was used in the classification modules because the primary label and the emotions may depend on either the title or the body. For span extraction, the system also kept the original text fields, since every predicted span had to appear literally in the input.

A simplified example of the input and output structure is shown below:


```
[
  {
    "row_id": 4311,
    "lang": "ES",
    "title": "Avance en mi tesis: superando la ansiedad inicial",
    "selftext": "Después de semanas paralizado por el miedo a no ser lo suficientemente competente,
        finalmente he comenzado a escribir mi tesis de grado. Al principio estaba aterrado de cometer
         errores irreversibles en mi investigación, pero he descubierto que cada capítulo que
        completo me da más confianza. Aunque sigo teniendo momentos de pánico sobre si mi análisis
        será aceptado, siento que las cosas van mejorando. Mi asesor me ha apoyado mucho y eso me ha
        ayudado a confiar en que podré terminar este proyecto con éxito.",
    "primary_label": "General Hope",
    "span_annotations": [
      {
        "span": "siento que las cosas van mejorando",
        "outcome_stance": "Desired",
        "actor": "World/System"
      }
    ],
    "trigger_emotions": [
      "fear",
      "sadness",
      "joy"
    ]
  },
  {
    "row_id": 1603,
    "lang": "EN",
    "title": "My best friend keeps ditching me for their new relationship",
    "selftext": "I've been friends with this person for 8 years and suddenly they're in a
        relationship and I don't exist anymore. We used to hang out constantly, now I'm lucky if I
        hear from them once a month. I'm genuinely terrified I'm going to lose them completely. But
        hey, maybe they'll finally realize how much they miss our friendship and come crawling back
        once the honeymoon phase ends and their partner inevitably gets tired of them. I'm sure that'
        s definitely going to happen and everything will magically go back to normal.",
    "primary_label": "Sarcastic Hope",
    "span_annotations": [
      {
        "span": "maybe they'll finally realize how much they miss our friendship and come crawling
            back once the honeymoon phase ends and their partner inevitably gets tired of them",
        "outcome_stance": "Desired",
        "actor": "Other"
      }
    ],
    "trigger_emotions": [
      "fear"
    ]
  }
]
```


As shown in Table 1, the corpus is bilingual and relatively balanced between Spanish and English. However, the labels are imbalanced: `Not Hope` is the most frequent class, while `Hopelessness` is the least frequent. For emotions, `sadness` and `anger` are more frequent than `joy`, `love`, and `surprise`.

Table 1
Compact summary of the corpus and training distributions.

Languages

Language	Train	Test
ES	2614	1087
EN	2243	995

Emotions

Emotion	Freq.
sadness	2438
anger	1335
Nuetral/unclear	1277
fear	1119
joy	614
love	611
surprise	389

Primary label

Label	Freq.
Not Hope	1400
General Hope	937
Realistic Hope	701
Sarcastic Hope	700
Unrealistic Hope	700
Hopelessness	419

Spans per example

Spans	Examples
0	1820
1	2631
2	236
3	99
4 or more	71

The span distribution is also important for the design of the pipeline. In Task C, 1,820 training examples have no annotated spans. These cases correspond mainly to posts where no outcome should be extracted, such as `Not Hope` or `Hopelessness`. In contrast, 3,037 examples have at least one span. Some posts contain more than three annotated spans, but the shared task requires systems to output at most three. For that reason, the post-processing stage ranks the predicted spans and keeps only the three best candidates.

Although the corpus consists of social media posts, the texts are not always short. In the training set, the mean length was 202.55 words, the median was 161, and the maximum was 5,912. This motivated the use of long enough input lengths for classification and a window-based strategy for span extraction.

4. Methodology

The methodology was organized as a modular text-processing pipeline. This decision follows from a central observation: the four outputs required by HOPE-EXP do not belong to a single family of problems. Tasks A and B operate over the complete post; Task C requires locating exact fragments within the sequence; and Task D semantically interprets each span. Therefore, the system was designed as a set of specialized modules that share the same base textual representation but produce different outputs.

4.1. Preprocessing and Input Representation

The first stage consisted of transforming each post into a textual representation suitable for the different subtasks. For Tasks A and B, `full_text` was created by concatenating `title` and `selftext`. This reduced the risk of losing information, since some posts expressed the relevant signal in the title, while others did so in the main body. The field `selftext` was interpreted as the body of the post.

No aggressive text cleaning was applied. Capitalization, punctuation, emojis, hashtags, and informal expressions were preserved because these elements may provide useful signals for sarcasm, emotion, or stance. Normalization was limited to converting empty values into empty strings and removing surrounding spaces. In all modules, the official label names were preserved, including `Nuetral/unclear`, to avoid output format inconsistencies.

Long contexts were handled differently depending on the subtask. For Tasks A and B, the model

received `full_text` with a maximum sequence length of 512 tokens. For Task C, span extraction was performed with overlapping windows and a stride, so that long posts could be processed without discarding later parts of the text. For Task D, the model did not receive the full post in all cases; instead, it received a local context window around the extracted span. This reduced computational cost and helped the model focus on the span, but it also introduced a possible loss of broader context, especially for actor classification.

4.2. General Pipeline Design

The final pipeline followed a sequential flow. First, Task A predicts `primary_label`. Then, Task B predicts the list of emotions. Task C is executed only when the primary label belongs to a hope category; if Task A predicts *Not Hope* or *Hopelessness*, the `span_annotations` list is directly set to empty. Finally, for each span produced by Task C, Task D predicts `outcome_stance` and `actor`. This structure acts as a *gating* mechanism: an early decision controls whether later stages should produce spans or not.

The modular design also allowed error analysis by subtask. For example, an incorrect Task A prediction may prevent Task C from extracting spans; an incomplete span from Task C may affect Task D classification; and a Task B error does not necessarily affect the remaining outputs, but it does modify the emotional component of the JSON output.

4.3. Task A: Primary Label Classification

Task A was formulated as a multi-class classification problem at the post level: each post had to receive exactly one label among *General Hope*, *Realistic Hope*, *Unrealistic Hope*, *Sarcastic Hope*, *Hopelessness*, and *Not Hope*. This subtask was developed first because its output conditions later stages of the pipeline; in particular, when the system predicts *Not Hope* or *Hopelessness*, no spans or semantic roles are generated.

As a starting point, a classic baseline was trained using TF-IDF and logistic regression with unigrams and bigrams. TF-IDF represents each post as a vector of weighted words, while logistic regression acts as a linear classifier that estimates the probability of each class. This baseline was useful for obtaining a simple reference before using more expensive models. In addition, it made it possible to detect early confusions between closely related labels, especially *Realistic Hope*, *Hopelessness*, and *Not Hope*. The use of a classic baseline also follows a common experimental logic in shared text-analysis tasks: starting with simple and interpretable methods and then comparing their performance against more complex contextual models. This type of comparison appears in related work on hope speech and sarcasm detection, where systems based on traditional representations serve as references for neural or transformer-based models [10, 11].

Several multilingual transformer models were then evaluated: `mBERT`, `twitter-xlm-roberta-base`, `xlm-roberta-base`, and `xlm-roberta-large`. This comparison was performed because the corpus contains Spanish and English social media texts. Prior work on HOPE and PolyHope shows that hope-related categories, such as expectation, optimism, sarcasm, and hopelessness, can be closely related in social media; therefore, it was necessary to test models capable of representing context rather than relying only on word frequency [4, 2, 5, 6]. `twitter-xlm-roberta-base` was included because it is oriented toward social media text, although it did not outperform XLM-R in our experiments.

The final model selected was `xlm-roberta-large`, fine-tuned using `full_text` as input. It was chosen because it obtained the best validation Macro-F1 and showed the most stable behavior across classes and languages. Macro-F1 was important because it averages performance across classes and prevents the result from being dominated by *Not Hope*, the most frequent class.

4.4. Task B: Multi-label Emotion Detection

Task B was formulated as a multi-label classification task. Each example was encoded as a binary vector with seven positions, one per emotion. Unlike Task A, the outputs are not mutually exclusive: a post may express, for example, sadness and fear at the same time. Therefore, the model used an independent output for each emotion and a loss function suitable for multi-label classification.

This formulation is related to recent work on text-based emotion detection, especially in multilingual and multi-label scenarios. For example, in SemEval-2025 Task 11, emotion detection is formulated as a problem where the same text can be associated with multiple emotions simultaneously [12]. In addition, work on emotion semantic parsing shows that emotions can be linked to different semantic constituents, such as triggers, entities, or contextual relations [13]. Although Task B does not perform full emotion semantic analysis, this background supports the decision to treat emotions as multiple signals associated with the textual content.

Models based on XLM-R and mDeBERTa were evaluated. The best architecture was `xlm-roberta-large` with a weighted loss using `pos_weight`. This weighting was incorporated because the emotions were imbalanced: `sadness` appeared much more frequently than `surprise`, `love`, or `joy`. The goal was to reduce the model's bias toward the most frequent emotions and increase sensitivity to minority classes.

In addition to model training, a specific threshold calibration stage was carried out. In multi-label classification, probabilities are converted into labels using a decision threshold. First, the global threshold 0.5 was tested; then, a threshold was searched for each emotion; finally, thresholds were estimated by emotion and language. This last configuration produced the best internal result, although the improvement over emotion-only thresholds was small. During inference, if no emotion exceeded its threshold, the emotion with the highest probability was selected to avoid empty `trigger_emotions` lists.

The Task A prediction was also tested as an auxiliary signal for Task B. The idea was to provide the model with a hint about the type of hope before predicting emotions. However, this variant did not outperform the best version based on `xlm-roberta-large` and thresholds calibrated by emotion and language, so it was kept as a comparative experiment but not as the final configuration.

4.5. Task C: Span Extraction

Task C was addressed as an information extraction task. Unlike Tasks A and B, assigning a global label is not enough: the system must locate exact text fragments that represent expected or avoided outcomes. For this reason, the initial methods based on rules and sentence classification were used only as baselines. The rules searched for explicit patterns such as *hope*, *wish*, *ojalá*, or *espero que*; however, they showed low recall. Sentence classification also did not work well, because outcomes often appear as clauses or internal fragments, not as complete sentences.

The initial exploration with rules and textual candidates was inspired by work on emotion cause extraction in social media, where triggers or causes are modeled as localized fragments within the text [14]. However, in HOPE-EXP, the span must be an exact substring of the input. This requirement made it necessary to move from general rules to a more precise sequence labeling approach.

The final solution formulated Task C as a token-level sequence labeling problem. `xlm-roberta-large` was used with a BIO scheme composed of three labels: `O`, `B-SPAN`, and `I-SPAN`. To build the training labels, the gold spans were aligned with the tokenizer offsets. This alignment made it possible to mark which sub-tokens belonged to the span and later reconstruct the prediction as a substring of the original text.

During inference, the model produced token-level labels and associated probabilities. For long posts, the text was processed with overlapping windows, using a maximum length of 512 tokens and a stride. This avoided truncating the post after the first 512 tokens. BIO labels were then grouped to form span candidates and reconstructed using the original character offsets.

Post-processing was a central part of this subtask. Very short or uninformative spans were removed, external punctuation was trimmed, nearby spans were merged, duplicates were removed, and the output was limited to at most three spans. This last step was necessary because the dataset contains some training examples with more than three annotated spans, while the official output schema allows only up to three. A BIO+CRF variant was tested to model dependencies between consecutive labels, but it did not improve over the BIO model with post-processing.

4.6. Task D: Stance and Actor Classification

Task D was formulated as span-conditioned classification. To train the models, the dataset was transformed from one row per post into one row per span, keeping `row_id`, `lang`, `primary_label`, `full_text`, span, `outcome_stance`, and `actor`. This transformation allowed each training example to correspond directly to a semantic unit that had to be interpreted.

This decision is related to the idea that emotions and expectations do not only appear as global labels, but may also be associated with specific textual constituents and with the entities or actors involved [13]. Therefore, Task D used not only the complete post, but also the span as the central unit of interpretation.

First, a multitask model with two classification heads was trained: one for `outcome_stance` and another for `actor`. The loss combined both outputs, and class weights were used to handle imbalance. The results showed that `outcome_stance` was easier than `actor`: distinguishing *Desired* from *Avoided* depended mainly on the meaning of the outcome, while identifying the actor required inferring who was linked to the result, sometimes implicitly.

A span-aware approach was then tested. In this variant, the target span was explicitly marked inside the context using special tokens:

```
<SPAN> target span </SPAN>
```

The model combined a global representation of the input with a specific representation of the tokens inside the marked span. This strategy improved `actor`, because it forced the model to focus on the fragment that had to be interpreted. However, it also exposed a limitation of the pipeline: once a span is extracted, the role classifier may receive less discourse context than the original post contained. This can affect actor identification, since the actor is sometimes implicit or appears outside the extracted span.

The final Task D setup used separate models for `outcome_stance` and `actor`, with a three-fold out-of-fold evaluation. The configuration used `xlm-roberta-large`, a context window around the span, a maximum length of 256 tokens, partial layer freezing, weighted loss, and light oversampling for minority classes. This configuration sought to balance performance and computational cost. Heavier variants, such as 5-fold training without frozen layers or a maximum length of 512, were discarded because they substantially increased training time and memory usage without guaranteeing proportional gains.

4.7. Final Integration and JSONL Validation

The final pipeline generated the *submission.jsonl* file from the test set. The integration applied the following rules: `primary_label` comes from Task A; `trigger_emotions` comes from Task B; if `primary_label` is *Not Hope* or *Hopelessness*, `span_annotations` is set to an empty list; if it corresponds to a hope category, Task C extracts up to three spans; and, for each span, Task D predicts `outcome_stance` and `actor`.

Before saving the submission, a structural validation of the JSONL file was applied. The system checked that each `row_id` was present and unique, that all labels belonged to the official label sets, that `trigger_emotions` and `span_annotations` were lists, that there were no more than three spans per example, and that each span appeared literally in the text. This last validation was especially important for Task C, since models may generate reasonable fragments that are not necessarily exact.

5. Experimental Setup

Internal evaluation was mainly performed using train-validation splits, with an 80/20 proportion for the first subtasks. For Task D, a three-fold out-of-fold evaluation was also used, because a more robust estimate over all available spans was needed.

Table 2
Internal results for model selection in Tasks A, B, and C.

Task	Configuration / purpose	Macro/ROUGE	Micro
A	TF-IDF + Logistic Regression; classic baseline	0.8098	–
A	Balanced TF-IDF; imbalance analysis	0.8063	–
A	mBERT; multilingual transformer reference	0.8644	–
A	Twitter-XLM-R; social-media-oriented model	0.8570	–
A	XLM-R base; multilingual contextual model	0.9128	–
A	XLM-R large; final Task A model	**0.9563**	–
B	XLM-R base, threshold 0.5	0.6898	0.7190
B	XLM-R base, emotion thresholds	0.7119	0.7359
B	XLM-R large, emotion thresholds	0.7575	0.7715
B	XLM-R large, emotion+language thresholds	**0.7579**	**0.7721**
B	mDeBERTa-v3-base	0.3554	0.3736
B	XLM-R large + Task A signal	0.7472	0.7643
C	Rules	0.5238	–
C	Sentence classification	0.3745	–
C	BIO with XLM-R large	0.7662	–
C	BIO + post-processing	**0.8324**	–
C	BIO + CRF	0.7512	–

The metrics used were Macro-F1 for classification tasks, Micro-F1 as a complementary metric in Task B, and approximate ROUGE-1 for Task C. In Task D, Macro-F1 was reported separately for `outcome_stance` and `actor`. Since some metrics come from internal evaluations with different validation schemes, the results should not be interpreted as the official competition result, but rather as experimental evidence for model selection.

6. Results

Table 2 summarizes the internal results used to select the models for Tasks A, B, and C. Multilingual contextual models outperformed classic baselines in Task A. For Task B, the best version was `xlm-roberta-large` with weighted loss and thresholds calibrated by emotion and language. For Task C, the BIO-based solution with post-processing was the best, showing that span extraction requires working at the token level and carefully reconstructing the final fragments.

Table 3 summarizes the main Task D experiments and the final pipeline configuration. The best version on the fixed 80/20 split was the hybrid strategy with an actor-only ensemble. The most robust global evaluation was the lightweight k-fold approach with separate models for `outcome_stance` and `actor`. In the k-fold setting, `outcome_stance` obtained a Macro-F1 of 0.9020, while `actor` obtained 0.6838. This difference confirms that `actor` was the most difficult part of Task D. In particular, *Unclear* reached an F1 of 0.40 in the OOF evaluation, although the system correctly identified 47 of 111 real cases of this class.

The approximate internal average of the five main metrics was 0.8225. This value should be interpreted with caution because it comes from internal validations and not from the official test set.

7. Error Analysis

The experimental analysis identified several types of errors. In Task A, the classic models more frequently confused semantically close classes, especially *Realistic Hope*, *Hopelessness*, and *Not Hope*. This suggests that the difficulty was not only in detecting hope-related vocabulary, but also in distinguishing whether the text expresses a concrete expectation, an absence of hope, or a description without future-oriented intent.

Table 3
Task D results and internal summary of the final pipeline.

Model / component	F1 stance	F1 actor	Average
Multitask	0.8361	0.6280	0.7321
Span-aware multitask	0.8204	0.6542	0.7373
Hybrid + ensemble	0.8361	**0.7126**	0.7743
K-fold light OOF	**0.9020**	0.6838	**0.7929**
Final Task A	Macro-F1		0.9563
Final Task B	Macro-F1		0.7579
Final Task C	ROUGE-1		0.8324
Final Task D stance	Macro-F1		0.9020
Final Task D actor	Macro-F1		0.6838
Approximate internal average	–		0.8225

In Task B, the main challenge was the imbalance between emotions and the multi-label nature of the problem. Less frequent emotions, such as `joy`, `love`, and `surprise`, required specific threshold calibration. A global threshold adjustment was not sufficient, whereas thresholds by emotion and language produced better results.

In Task C, the rule-based approach showed low recall, and sentence classification did not recover positive spans. The main problem was that outcomes do not always correspond to complete sentences; they often appear as clauses or internal fragments. The BIO model handled this localization better, but it required post-processing to clean boundaries, merge nearby fragments, and remove uninformative spans.

In Task D, the main bottleneck was `actor`. The *Unclear* class was especially difficult because it can depend on implicit information or pragmatic interpretation of the text. In addition, actor classification is strongly connected to the previous stages of the pipeline. If Task C extracts a span that is too short or lacks surrounding context, the role classifier may lose information needed to determine who is associated with the outcome. This explains why `outcome_stance` achieved higher performance than `actor`: stance is often expressed inside the span itself, whereas the actor may be located elsewhere in the post or inferred from broader discourse.

8. Conclusions

This work presented a modular solution for HOPE-EXP 2026. The system addresses outcome-oriented hope analysis through four components: primary label classification, multi-label emotion detection, span extraction, and semantic classification of stance and actor. The main design decision was to specialize the system by subtask instead of using a single model for the entire challenge.

The internal results show that multilingual transformer models, especially `xlm-roberta-large`, were effective for classification and extraction tasks. Task A achieved high and stable performance. Task B improved through weighted loss and threshold calibration. Task C required a BIO sequence-labeling formulation and careful post-processing. Task D showed that separating `outcome_stance` and `actor` was more effective than using a single multitask model.

The main limitation of the system was `actor` classification, especially for the *Unclear* class. This limitation is related not only to class imbalance, but also to the pipeline structure: once spans are extracted, some broader contextual information may be lost before role classification. Even with this limitation, the official result shows that a controlled modular transformer-based pipeline can remain competitive with more general large language model approaches.

As future work, it would be useful to explore context-preserving role classification, where Task D receives both the extracted span and a richer representation of the full post. It would also be useful to test controlled normalization for social media texts. During corpus analysis, several texts were

observed to contain links, emojis, repetitions, abbreviations, and intentionally non-standard spelling. Although this work avoided aggressive cleaning in order not to remove useful signals for emotion or sarcasm, a more controlled normalization stage could improve the text representation without losing relevant information.

Declaration on Generative AI

During the preparation of this manuscript, generative AI tools were used to support text organization, language editing, and translation into English. These tools were not used to generate experimental results, modify the evaluation metrics, or make methodological decisions. The technical content, experiments, results, and conclusions were reviewed and validated by the authors. All methodological decisions and analyses presented in this paper are the responsibility of the authors.

References

[1] 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).

[2] 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.

[3] 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.

[4] D. García-Baena, F. Balouchzahi, S. Butt, M. Á. García-Cumbreras, A. L. Tonja, J. A. García-Díaz, 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.

[5] F. Balouchzahi, G. Sidorov, A. Gelbukh, PolyHope: Two-Level Hope Speech Detection from Tweets, Expert Systems with Applications 225 (2023) 120078.

[6] 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.

[7] 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.

[8] S. Butt, F. Balouchzahi, A. I. Amjad, M. Amjad, H. G. Ceballos, S. M. Jiménez-Zafra, Optimism, Expectation, or Sarcasm? Multi-Class Hope Speech Detection in Spanish and English, arXiv preprint arXiv:2504.17974, 2025.

[9] 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.

[10] P. K. Kumaresan, B. R. Chakravarthi, S. C. Navaneethakrishnan, M. Á. García-Cumbreras, S. M. Jiménez-Zafra, J. A. García-Díaz, R. Valencia-García, M. Hardalov, I. Koychev, P. Nakov, D. García-Baena, K. K. Ponnusamy, B. Preston, Overview of the Shared Task on Hope Speech Detection for Equality, Diversity, and Inclusion, in: Proceedings of the Third Workshop on Language Technology for Equality, Diversity and Inclusion, Varna, Bulgaria, 2023, pp. 47–53. doi:10.26615/978-954-452-084-7_007.

[11] I. Abu Farha, S. V. Oprea, S. R. Wilson, W. Magdy, SemEval-2022 Task 6: iSarcasmEval, Intended Sarcasm Detection in English and Arabic, in: Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), Association for Computational Linguistics, 2022, pp. 802–814. URL: https://aclanthology.org/2022.semeval-1.111/.

[12] S. H. Muhammad, N. Ousidhoum, I. Abdulmumin, S. M. Yimam, J. P. Wahle, T. Ruas, M. Beloucif,

C. De Kock, T. D. Belay, I. S. Ahmad, N. Surange, D. Teodorescu, D. I. Adelani, A. F. Aji, F. Ali, V. Araujo, A. A. Ayele, O. Ignat, A. Panchenko, Y. Zhou, S. M. Mohammad, SemEval-2025 Task 11: Bridging the Gap in Text-Based Emotion Detection, in: Proceedings of the 19th International Workshop on Semantic Evaluation (SemEval-2025), Association for Computational Linguistics, 2025, pp. 2558–2569. URL: https://aclanthology.org/2025.semeval-1.327/.

[13] X. Jiang, Z. Wang, G. Zhou, End-to-End Emotion Semantic Parsing, in: Findings of the Association for Computational Linguistics: ACL 2024, Association for Computational Linguistics, 2024, pp. 37–47. URL: https://aclanthology.org/2024.findings-acl.4/.

[14] D. Xiao, R. Xia, J. Yu, Emotion Cause Extraction on Social Media without Human Annotation, in: Findings of the Association for Computational Linguistics: ACL 2023, Association for Computational Linguistics, 2023, pp. 1455–1468. URL: https://aclanthology.org/2023.findings-acl.94/.