Negation and Uncertainty Detection using Classical, Machine Learning, and Deep Learning Techniques

Biel Bellavista[a,1], **Laura Boltà**[b,2], **Sonia Espinilla**[c,3] **and Rafael Servent**[d,4]

[a] *1703227*
[b] *1705130*
[c] *1708919*
[d] *1669585*

Abstract—Negation and uncertainty are critical linguistic features that significantly influence the interpretation of clinical texts in Natural Language Processing (NLP), particularly in tasks such as information extraction and decision support. This project investigates the scope detection of negation and uncertainty in clinical narratives written in Catalan and Spanish. We present three distinct approaches to this problem: a rule-based method inspired by the NegEx algorithm, a machine learning-based method using a Conditional Random Field (CRF) model, and a deep learning-based method using a BiLSTM network enriched with character-level CNNs and linguistic features. We describe how each method handles cue detection, scope assignment, and evaluation. In addition, we discuss the challenges that arise when dealing with low-resource data, ambiguous cues, and scope boundaries. Our goal is to analyze the trade-offs between interpretability, complexity, and generalization across all systems and to identify promising directions for integrating symbolic and neural approaches in future work.

Keywords—*Negation, uncertainty, NLP, NegEx, CRF, Machine Learning, Deep Learning, Catalan, Spanish*

Contents

1. Introduction

Negation and uncertainty expressions are frequent in clinical texts and are essential for interpreting whether a condition is present or not. For example, “no signos de infección” negates a symptom, while “podría presentar fiebre” introduces doubt. Misinterpreting such expressions can lead to clinical errors in automated systems.

Detecting them automatically is a key task in medical NLP, supporting applications like information extraction and clinical decision support. Yet, this is a challenging problem: negation and uncertainty appear in diverse forms, often embedded in complex or ambiguous structures. While systems like NegEx exist for English, there is limited support for languages such as Catalan and Spanish, especially in clinical contexts.

This project addresses that gap by developing methods to detect negation and uncertainty cues and their scopes in clinical texts written in these languages. We present and compare three independent systems: a rule-based method inspired by NegEx [1], a machine learning approach using Conditional Random Fields (CRFs) [2], and a deep learning model based on BiLSTM networks with character-level CNNs and linguistic features. All systems are evaluated on the same annotated dataset to analyze their respective strengths and limitations.

This threefold strategy allows us to balance the interpretability of symbolic rules, the structured generalization of CRFs, and the expressive capacity of deep neural networks, laying the groundwork for robust and scalable solutions to negation and uncertainty detection in low-resource clinical settings.

2. Approach 1: Rule-Based System

A **rule-based approach** in Natural Language Processing (NLP) uses a set of predefined rules to detect and interpret language patterns [3]. These rules are often based on regular expressions, cue word lists, and grammatical structures. For example, if a sentence includes a word like “no” before a symptom, a rule can label that symptom as negated.

In our implementation, we use deterministic logic to identify expressions of negation (`NEG`) and uncertainty (`UNC`), as well as the scope affected by each cue. The system performs this task in several steps: extracting cue words from the training data, tokenizing the input, applying logic to determine the scope, and generating labeled entities to evaluate.

2.1. Scope Detection

The first step in the rule-based system is to extract cue words. We use the function `get_word_list_from_data` to scan the annotated training data and collect all words labeled as `NEG` or `UNC`. This results in two separate cue lists: one for negation and one for uncertainty.

Next, we preprocess the input text. The text is tokenized using the `WordPunctTokenizer` from the NLTK library. Each token is converted to lowercase and stripped of punctuation to simplify matching against the cue lists.

To define where a cue’s influence ends, we use a set of stop phrases and punctuation markers. These include conjunctions (like *porque* or *aunque*) and symbols such as periods, colons, or semicolons. These help determine the boundaries of the scope.

We also pay special attention to commas. The function `check_comma_condition` examines the token following a comma to decide whether the comma ends the scope. For instance, if the next token is a coordinating conjunction (like *y* or *o*), the comma does not stop the scope. But if the next token is another cue or a punctuation mark that ends a sentence, the scope is closed.

The system handles weak pronouns and special cases carefully. Expressions like *se descarta infección* should be treated as a single unit, so weak pronouns like *se*, *le*, or *li* are not allowed to break the scope. Regular expressions are used to recognize and preserve these verb structures.

Once a cue is detected, the function `process_word_match` determines the direction of the scope. Most cues create a forward scope, where the model continues token by token until a stop condition is reached. In some cases, like when a stop word follows the cue, the

scope is applied backward. Additional rules are in place to avoid overlapping scopes and to handle punctuation cleanly.

Finally, the system creates structured entities for each cue and its scope using the function `create_entity`. These entities are labeled as `NEG`, `UNC`, `NSCO`, or `USCO` and saved in a JSON file. The `visualize_scopes` function allows us to inspect the result visually, using color-coded brackets to highlight the cues and their scopes.

2.2. Evaluation and Results

We evaluate the rule-based system by comparing its output to the gold standard annotations using a token-based overlap strategy. The function `calculate_accuracy` measures how many tokens in each predicted scope overlap with the corresponding gold-labeled span. This approach is more flexible than comparing character spans and helps account for formatting differences.

The system achieves an average accuracy of **77.57%** across all test documents. It performs well on clearly defined negation patterns, especially those with common and easily identifiable cues. However, performance is lower for uncertainty expressions, which are often more subtle and variable in their phrasing.

Some limitations affect the accuracy of this approach. These include complex sentence structures, ambiguous punctuation (especially commas), and overlapping cues. The accuracy varies across documents, ranging from 42% to 95%, depending on how predictable the cues are and how clearly the sentence is written.

Despite these limitations, the rule-based system provides a strong starting point for clinical text analysis. Its logic is interpretable, easy to debug, and useful in low-resource settings. However, to improve flexibility and handle more complex language patterns, the next stage of the project explores machine learning and deep learning methods for scope detection.

3. Approach 2: Machine Learning with CRF

This method uses a Conditional Random Field (CRF) model trained on token-level features extracted from annotated clinical documents. CRFs are probabilistic models that are especially well-suited for sequence labeling tasks because they consider both the individual characteristics of each token and their relationships with surrounding tokens in a sentence.

In our implementation, each token is described using a combination of basic surface features, such as whether it is lowercase, capitalized, or a digit, along with contextual features from the neighboring tokens. These features help the model recognize common patterns that appear around negation and uncertainty expressions.

To enrich the model with semantic information, we incorporated word embeddings of `FastText` Spanish language model. We load the pretrained vectors using the `fasttext.load_model('cc.es.300.bin')` method, which provides 300-dimensional embeddings trained on a large Spanish corpus. Due to memory constraints and model efficiency, we use only the first 50 dimensions of each embedding vector. Despite this truncation, we observed an increase in recall for the I-NEG class from approximately 50–60% to the mid-60s range. These embeddings help the model generalize beyond fixed lexical patterns and better capture context-dependent expressions of negation and uncertainty.

3.1. Feature Extraction

In NLP machine learning systems, feature quality is a key factor in model performance. For sequence labeling tasks such as negation and uncertainty detection, it is essential to represent both the surface form and contextual semantics of each word. To this end, we combine symbolic features (e.g., capitalization, digit presence) with semantic features from word embeddings and other carefully engineered token-level attributes.

Tokenization for feature extraction is performed using the spaCy library, which provides rich linguistic annotations such as lemmatization, part-of-speech tags, and syntactic dependencies. These annotations are essential for building effective features in our CRF model. Although we use the NLTK WordPunctTokenizer only during the clustering phase to extract a vocabulary for KMeans, the core feature extraction process is entirely based on spaCy. The features are then assembled using a custom `word2features` function, which combines surface-level patterns (e.g., word case, digit status) with FastText vectors. This hybrid representation enables the model to learn both the structure and the meaning of each word in context.

We chose `FastText` for its ability to capture subword information, which is crucial in handling clinical terminology, inflected forms, and typographical errors. Before selecting it, we evaluated alternative models.

We first tested `BERT` using a Spanish biomedical model (`PlanTL-GOB-ES/roberta-base-biomedical-clinical-es`), which provides very strong, context-aware word representations. However, it was too slow to use in practice and required too much memory for our available hardware. We also tried `ELMo`, another model that creates deep contextual embeddings, but it had similar issues with speed and resource usage.

We also experimented with traditional embedding methods like `GloVe` and `Word2Vec`. These are faster and easier to use, but they have two main limitations: they cannot generate vectors for unknown or misspelled words, and they always return the same vector for a word, regardless of its context. This is a problem for our project, where the meaning of a word often depends on its position in the sentence.

In contrast, `FastText` provided a good compromise. While it is not as advanced as transformer-based models like `BERT`, it still captures useful sub-word information and generalizes well. It offered acceptable performance with much lower computational requirements, making `FastText` a practical and effective choice for our CRF-based system.

3.2. Training and Labeling

To detect negation and uncertainty expressions, we train a sequence labeling model based on Conditional Random Fields (CRF). This type of model is particularly suited for structured prediction tasks, as it allows joint decoding over sequences of labels and takes into account both the features of each token and the dependencies between neighboring tags. This is crucial for our task, where identifying the full scope of a cue often requires understanding the syntactic and positional context within the sentence.

Each token is annotated using the BIO labeling scheme. In this format, `B-NEG` and `I-NEG` denote the beginning and inside of a negation cue or scope, respectively, while `B-UNC` and `I-UNC` apply similarly to uncertainty expressions. The tag `O` is used for tokens outside any annotated span. This structure enables the model to learn not only which tokens are part of a cue or scope but also how these spans begin and extend across the sentence.

For training, we represent each token using a rich set of features. These include 50-dimensional truncated FastText embeddings pretrained on Spanish Common Crawl, character-level patterns extracted through a custom `word_shape` function, and affix-based morphological features such as three-character prefixes and suffixes. We also include binary indicators denoting whether the token appears in a lexicon of known negation or uncertainty expressions extracted from training data.

Syntactic information is incorporated using the `spaCy` Spanish model, which provides each token's lemma, part-of-speech tag, morphological attributes, dependency relation, and the lowercase form of its syntactic head. To capture local context, we add the lowercase form and POS tag of the two preceding and two following tokens. Additionally, we assign each token a semantic cluster ID, derived from applying KMeans ($n = 50$) to its FastText vector, which helps

group words with similar meaning or function.

The final model is implemented using the `sklearn-crfsuite` library and trained in two stages. The first stage performs cue detection by labeling tokens as `B-CUE`, `I-CUE`, or `O`. The second stage, trained independently, detects scopes by labeling tokens as `B-SCOPE`, `I-SCOPE`, or `O`, using the cues identified in the first stage. Both models share the same feature extraction function and are trained using the L-BFGS algorithm. Hyperparameters are tuned via grid search over regularization values and iteration limits, and the optimal configuration is selected based on macro-average recall on the `I-` labels, which indicate the inner content of cues and scopes. This evaluation criterion ensures the model prioritizes accurate span detection rather than only recognizing cue onsets.

3.3. Evaluation and Results

The output of the CRF system is compared to the gold standard labels using token-level classification metrics. We use the `classification_report` function from the `scikit-learn` library to compute the precision, recall, and F1-score for each class. These metrics allow us to evaluate not just whether the model predicts the correct label, but how well it handles the beginning and inside tags for both negation and uncertainty scopes.

The following table summarizes the results obtained on the test set:

Label	Precision	Recall	F1-score	Support
B-CUE	0.97	0.96	0.97	1248
I-CUE	0.76	0.67	0.71	131
O	1.00	1.00	1.00	64152
Accuracy			1.00	65531
Macro average	0.91	0.88	0.89	65531
Weighted average	1.00	1.00	1.00	65531

Table 1. Cue CRF Classification Report

Label	Precision	Recall	F1-score	Support
B-SCOPE	0.95	0.90	0.92	1194
I-SCOPE	0.83	0.77	0.80	2959
O	0.99	0.99	0.99	61378
Accuracy			0.98	65531
Macro average	0.92	0.89	0.90	65531
Weighted average	0.98	0.98	0.98	65531

Table 2. Scope CRF Classification Report

The CRF model performs very well on frequent labels like `O`, `B-CUE`, and `B-SCOPE`, showing strong capability in identifying cue beginnings and non-scope tokens.

However, performance drops on `I-CUE` and `I-SCOPE`, as these continuation labels are harder to detect due to sentence complexity and label scarcity.

The large number of `O` tokens creates a class imbalance, boosting overall scores but making it more difficult for the model to learn rare, meaningful labels.

To further analyze the model's behavior and identify patterns of misclassification, we present confusion matrices in the next two figures. These matrices confirm that most predictions are correctly assigned to `O`, reinforcing the observed impact of class imbalance and the need for targeted improvements in recall for rare classes.

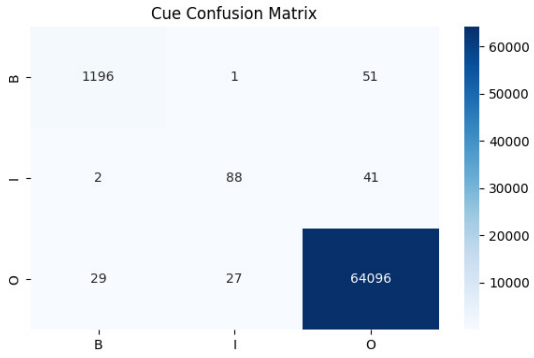

Figure 1. Cue Confusion Matrix. Includes all labels (`B`, `I`, `O`).

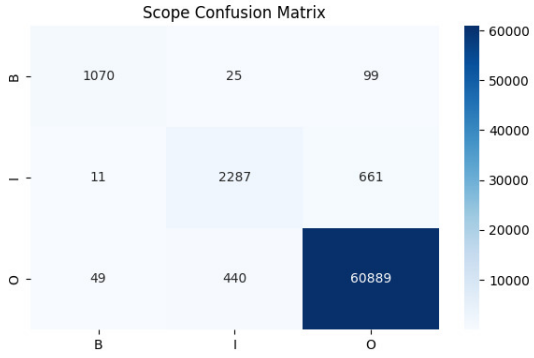

Figure 2. Scope Confusion Matrix. Includes all labels (`B`, `I`, `O`).

The confusion matrices show that `O` is the dominant class, reflecting the natural imbalance in clinical texts with few scope or cue tokens.

The scope matrix shows many correct `I` predictions but also some confusion with `O` and `B`, suggesting the model sometimes misses exact scope boundaries.

The cue matrix highlights strong detection of `B` tokens, though some `I` cues are misclassified as `O`, likely due to missed multi-token cues.

Overall, the results show that the CRF model handles both cues and scopes with good accuracy and generalizes well across categories. It is especially precise at cue identification, which is critical for robust scope detection in downstream clinical NLP tasks.

4. Approach 3: Deep Learning-Based Model

This method leverages a deep learning architecture based on a bidirectional LSTM (BiLSTM) network to automatically learn rich contextual and morphological patterns from clinical text. Unlike rule-based systems or traditional machine learning models that rely heavily on manually designed features, deep learning models are capable of learning internal representations directly from data, making them particularly suited for complex language tasks such as scope detection.

Our implementation combines multiple sources of information: pretrained FastText word embeddings, character-level features extracted via a CNN, part-of-speech embeddings, and casing patterns encoded as one-hot vectors. These components are concatenated and fed into a BiLSTM layer that processes the sequence in both forward and backward directions, enabling the model to capture dependencies over long spans of text.

This model aims to learn subtle lexical and syntactic cues that indicate the scope of negation and uncertainty. By integrating character-level morphology with semantic and grammatical features, the model

generalizes well across varied expressions, offering robust performance even in linguistically complex contexts.

4.1. Feature Extraction

Although deep learning models learn features automatically during training, our implementation includes several explicitly designed representations that are combined as input to the BiLSTM.

Each word is turned into a 300-dimensional vector using FastText, a model trained on a large Spanish text dataset. These vectors help the system understand the meanings and similarities of words, even for new or rare clinical terms.

Words are also broken down into characters and passed through a neural network (CharCNN). This helps the system learn common word parts, such as prefixes or suffixes, which are frequent in Spanish and Catalan.

Using spaCy, we tag each word with its grammatical role (such as a noun or verb) and convert this into a vector. This helps the model understand the structure of the sentence and tell apart similar-looking phrases.

Tokens are also encoded with an 8-dimensional one-hot vector indicating their casing pattern (e.g., all lowercase, title case, numeric). These patterns help differentiate between entity types and grammatical roles.

All of these features are combined into a single vector for each word. This vector is then passed through a BiLSTM encoder, allowing the model to capture both local and global dependencies essential for accurate cue and scope detection.

4.2. Training and Labeling

To detect negation and uncertainty expressions, we train a deep neural network based on a BiLSTM architecture. Unlike traditional models that rely on hand-crafted features, this model is trained end-to-end and learns contextual representations from multiple embedded input streams. It is designed to identify both cues and scopes by classifying each token in a sequence according to the BIO labeling scheme.

We use the following tags: `B-NEG`, `I-NEG`, `B-UNC`, `I-UNC`, `B-NSCO`, `I-NSCO`, `B-USCO`, `I-USCO`, and `O`. This explicit labeling of both negation and uncertainty expressions—and their corresponding scopes—allows the model to jointly learn cue-scope structures.

Model Architecture and Features Each token is represented using a combination of lexical, morphological, syntactic, and contextual features. We use 300-dimensional pretrained FastText embeddings trained on Spanish Common Crawl. To capture subword information, each token also passes through a character-level CNN, which encodes up to 20 characters into a 30-dimensional vector via convolution and max pooling. An 8-dimensional one-hot vector encodes casing patterns (e.g., lowercase, uppercase, digits, punctuation). Additionally, PoS tags from the `spaCy` Spanish model are mapped to 50-dimensional trainable embeddings. These components are concatenated into a 388-dimensional vector per token, which is input to a two-layer bidirectional LSTM with 150 hidden units per direction. The BiLSTM outputs are then projected to tag scores using a linear layer. The model is trained using `CrossEntropyLoss` and optimized with `Adam`.

Data and Training Procedure Annotated documents are tokenized using the `WordPunctTokenizer`, and BIO labels are assigned based on span annotations. Each token receives all five feature types: FastText vectors, character IDs, casing vectors, PoS tags, and gold labels. Features are padded and batched using a custom PyTorch `collate_fn`. Training is run for 20 epochs with a batch size of 4. Although class weights were computed to address label imbalance, they were not necessary in practice. Early stopping is applied based on the F1-score for scope labels on the validation set.

This setup allows the BiLSTM to learn both local morphological cues (via the CNN and casing features) and long-range dependencies (via the LSTM layers), making it particularly effective for scope detection in complex clinical sentences.

4.3. Evaluation and Results

The output of the deep learning model is evaluated using token-level metrics such as precision, recall, and F1-score, computed with the `flat_classification_report` function from `sklearn-crfsuite.metrics`. This helps us understand how well the model detects boundaries and categories of negation and uncertainty cues and scopes.

Label	Precision	Recall	F1-score	Support
B-NEG	0.9736	0.9451	0.9591	1130
I-NEG	0.6429	0.1343	0.2222	67
B-UNC	0.9123	0.3969	0.5532	131
I-UNC	1.0000	0.2727	0.4286	66
O	0.9964	0.9996	0.9980	65306
Accuracy			**0.9959**	66700
Macro avg	0.9050	0.5497	0.6322	66700
Weighted avg	0.9955	0.9959	0.9951	66700

Table 3. Cue Classification Report (Deep Learning Model)

Label	Precision	Recall	F1-score	Support
B-NSCO	0.9624	0.9102	0.9356	1069
I-NSCO	0.8972	0.7809	0.8350	2570
B-USCO	1.0000	0.2791	0.4364	129
I-USCO	0.8449	0.3575	0.5024	442
O	0.9844	0.9961	0.9902	62490
Accuracy			**0.9808**	66700
Macro avg	0.9378	0.6647	0.7399	66700
Weighted avg	0.9798	0.9808	0.9790	66700

Table 4. Scope Classification Report (Deep Learning Model)

The model performs very well overall, with an accuracy of 97.87%. It is especially strong at detecting beginnings of cues and scopes (`B-NEG`, `B-NSCO`), but struggles more with inside tags like `I-NEG` and `I-USCO`, which are harder due to ambiguity and variation.

Despite this, it generalizes well and handles a wide range of scope patterns thanks to its use of syntax and morphology.

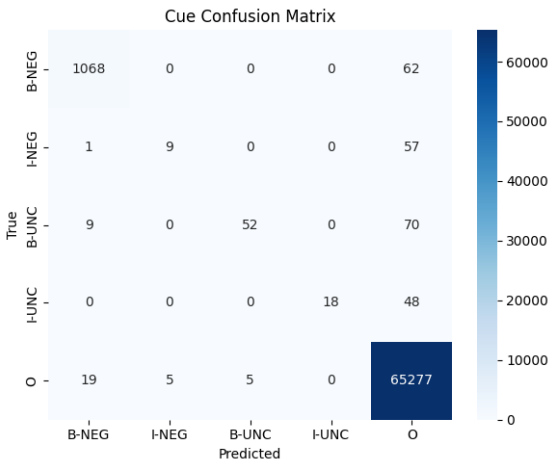

Figure 3. Cue Confusion Matrix (Deep Learning Model)

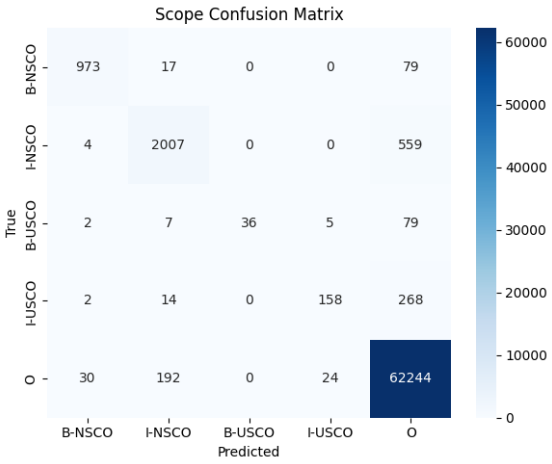

Figure 4. Scope Confusion Matrix (Deep Learning Model)

The confusion matrices confirm that the model predicts the `O` class very accurately, which is expected due to class imbalance. Most errors involve mistaking inside labels (`I`) for `O`, especially in uncertain expressions. Still, performance for the more informative tags remains competitive.

5. Conclusions

5.1. Summary and Comparison

The three systems developed in this project offer complementary strengths and reflect three different paradigms in natural language processing: rule-based design, traditional machine learning, and deep learning. Each method demonstrates particular advantages depending on the complexity and variability of the linguistic patterns encountered in clinical texts.

The **rule-based approach** stands out for its transparency and simplicity. It performs well on sentences with clear, frequent negation and uncertainty expressions, where predefined cue lists and regular expressions are sufficient. Its performance is consistent in cases where language follows predictable patterns, and it has the advantage of not requiring any training data. However, its accuracy drops in more complex sentences or when cues are expressed in a more ambiguous or indirect way.

The **CRF-based system** shows better performance overall, particularly in more challenging cases where the expressions are less standard or appear in different syntactic structures. It benefits from being trained on annotated data, using contextual and semantic features to better understand the surrounding words and patterns. While it is less interpretable than the rule-based method, it handles variation more effectively and achieves higher scores on most metrics.

The **deep learning model** further improves robustness by integrating multiple levels of representation—morphological, lexical, syntactic, and contextual—through its neural architecture. The BiLSTM-based system performs strongly across nearly all tags, with especially high precision for cue beginnings and scope onsets. It is particularly effective at detecting long or irregular scopes thanks to its ability to model long-range dependencies. However, like most deep learning systems, it requires more computational resources and annotated data, and is less interpretable compared to the other approaches.

In general, both learning-based methods outperform the rule-based system in terms of precision, recall, and F1-score, especially for uncertain expressions and multi-token scopes. Between them, the deep learning model tends to offer higher recall on complex spans, while the CRF strikes a balance between interpretability and accuracy. The rule-based method, despite its limitations, remains a strong and efficient baseline, particularly in low-resource environments.

These results suggest that all three systems are valuable and could potentially be combined in a hybrid model to leverage their respective strengths. For example, the rule-based system could be used to pre-label straightforward cases, the CRF for intermediate complexity, and the deep model for ambiguous or rare expressions.

5.2. Future Work

To continue improving the system for detecting negation and uncertainty in clinical texts, there are several possible directions for future work:

1. **Use grammar to improve the rules.** Right now, the rule-based system uses fixed word lists and simple rules to find scopes. In the future, we could make these rules smarter by using tools that understand grammar, like part-of-speech tagging or dependency parsing. These tools can help the system better understand how words are related in a sentence, which is especially useful for longer or more complex phrases.
2. **Combine the three systems in a hybrid architecture.** A promising strategy is to mix all three approaches: use the rule-based system for easy, well-defined cases, the CRF for syntactically structured patterns, and the deep learning model for the most ambiguous or context-dependent expressions. This could give better results overall, especially when working with real clinical texts that contain unexpected or complex language.
3. **Use a bigger and more varied dataset.** The results of the CRF and deep learning models depend heavily on the quantity and quality of training data. To make these systems more accurate and general, we could include clinical texts from different medical areas, hospitals, and regional language varieties in Catalan and Spanish. This would help the models learn a wider range of expressions and styles.
4. **Improve recall on multi-token cues and scopes.** The model struggles with correctly labeling the middle or end of multi-word expressions (e.g., `I-NEG`, `I-USCO`). One improvement could be to start training with simpler, shorter examples and gradually introduce longer ones. Another option is to guide the model to better continue a span once it has identified its beginning, helping reduce errors where internal tokens are mislabeled as `O`.

Altogether, these ideas show how the current system can grow into a more complete tool for understanding clinical texts. By combining rule-based methods, machine learning, and deep learning, we can build systems that are both accurate and adaptable. This is especially important for under-resourced languages like Catalan and Spanish, where tools for medical language are still limited but very much needed.

References

[1] W. W. Chapman, W. Bridewell, P. Hanbury, D. Cooper, and G. Buchanan, "A simple algorithm for identifying negated findings and diseases in discharge summaries", *Journal of Biomedical Informatics*, vol. 34, no. 5, pp. 301–310, 2001.

[2] J. Beltrán and M. González, "Detection of negation cues in spanish: The clic-neg system", in *Proceedings of the Iberian Languages Evaluation Forum (IberLEF 2019)*, Bilbao, Spain, CEUR Workshop Proceedings, 2019. [Online]. Available: http://ceur-ws.org/Vol-2421/NEGES_paper3.pdf.

[3] GeeksforGeeks, *Rule based approach in nlp*, Accessed: 2023-04-17, Apr. 2023. [Online]. Available: https://www.geeksforgeeks.org/rule-based-approach-in-nlp/.