Negation and Uncertainty Detection using Classical and Machine Learning Techniques

Pau González[a,1], Àlex Thomas[b,2], Pablo Necpas[c,3] and Rime Slaoui[d,4]

[a] *1708799*
[b] *1707555*
[c] *1706516*
[d] *1706847*

Abstract—Detecting negation and uncertainty in clinical texts is a crucial task for extracting reliable information from medical records. In this project, we tackle this problem by developing a rule-based system that identifies both the cue and the scope of negation and uncertainty expressions in texts written in Spanish and Catalan. Our current work focuses on implementing and evaluating a rule-based system. This serves as the foundation for future experiments using machine learning and deep learning, which will be explored in subsequent phases of the project.

Keywords—*Negation Detection, Clinical Text Mining, Rule-Based NLP, Uncertainty Analysis*

Contents

1. Introduction

Understanding whether a symptom or diagnosis is mentioned as present, absent, or uncertain in medical texts is essential for healthcare applications. This task, known as negation and uncertainty detection, enables the extraction of reliable information from clinical documents.

For example, if a medical note says “no presenta fiebre”, it is important that any system analyzing this text recognizes that the patient does **not** have a fever. Failing to detect this can lead to serious errors in decision-making.

In our case, **we focus on clinical narratives written in Spanish and Catalan,** where negation and uncertainty are expressed using various phrases that require specific handling.

We aim to create a system that combines keyword detection with syntactic analysis. **This report presents the first version of our system, focusing only on rule-based methods. In the future, we plan to integrate both machine learning and deep learning to improve coverage and accuracy.**

2. Data pre-processing

The goal of this section is to describe how the raw annotated data was prepared for the negation and uncertainty detection task. Several iterations were required to adapt our approach to the specificities of the data and improve the text quality prior to applying rule-based methods.

2.1. Dataset:

We worked with a large dataset of medical texts annotated with cues and scopes of negation (NEG”, NSCO”) and uncertainty (UNC”, USCO”). Each entry in the dataset is structured as a JSON object that contains the medical text and a list of manually annotated labels identifying the scope of negation and uncertainty in the document.

2.2. Text Cleaning:

To standardise and clean the text for downstream processing, we implemented multiple normalisation steps:

- **Lowercasing:**
  All characters were converted to lowercase to ensure case-insensitive matching.
- **Unicode Normalization:**
  Using `unicodedata.normalize(NFC’, text)` to ensure consistent character representations.
- **Whitespace Handling:**
  Multiple whitespaces were collapsed into a single space.
- **De-identification:**
  Patient identifiers marked by “*” were removed.
- **Punctuation Filtering:**
  Irrelevant punctuation was removed, keeping only medically relevant symbols.

```
def normalize_text(text):
    """Normalize medical text with Spanish/
    Catalan character support"""
    text = text.lower()
    text = unicodedata.normalize('NFC', text)
    text = re.sub(r'\s+', ' ', text).strip()
    text = re.sub(r'\*+', '', text)
    text = re.sub(r'[^\w\s.,;:!?-
                                 ]', '', text)
    return text

```

Code 1. Text-normalization

2.3. Language Handling:

As the dataset includes both Spanish and Catalan clinical texts, we applied language-aware preprocessing. Specifically, we used the spaCy natural language processing library with pre-trained medium-size models:

```

      es_core_news_md #for Spanish
      ca_core_news_md #for Catalan

```

These models provided reliable sentence and word tokenisation, part-of-speech tagging, and stopword filtering. In our preprocessing pipeline, we made sure to preserve all words that might be relevant for negation detection by retaining cue words even when they are typically considered stopwords.

```python
text = record.get("data", {}).get("text", "")
if not text:
    continue

results = analyze_medical_context(text, nlp)

```

Code 2. Text preprocessing and analisis

3. Rule-Based Implementation

This section describes the evolution of our rule-based approach for the given task. We iteratively refined our system based on progressively more complex data and linguistic insights.

3.1. Keyword-Based Detection

Our initial implementation was a **keyword-based model using regular expressions**. We defined two separate lists of cues: one for negation (e.g., *not, never, barely*) and another for speculation/uncertainty (e.g., *might, could, appears*). The function would scan a sentence and return whether it contained any of these cues.

This early model worked with small sample sentences in English to validate the regex logic and ensure the correct extraction of cues and their types. Although simple, this step allowed us to test token-level matching and build the structure of our output.

```python
def detect_negation_speculation(text):
    negations = negation_pattern.findall(text)
    speculations = speculation_pattern.findall(text)

    return {
        "contains_negation": bool(negations),
        "contains_speculation": bool(speculations),
        "negation_cues": negations,
        "speculation_cues": speculations
    }
```

Code 3. Basic regex-based negation/speculation detector

This simple logic was eventually replaced by a richer token-based system once we transitioned to real data.

3.2. Phrase Expansion

With access to a large dataset of Spanish and Catalan clinical texts, we designed a more sophisticated rule-based system that included: **custom tokenization** with character offsets; **prefix and suffix rules** for both negation and uncertainty cues; and **windowed scope expansion**, marking tokens following or preceding cues, unless interrupted by scope-breaking punctuation.

Scope-breaking tokens included period, semicolon, colon, etc., and were used to prevent cues from marking unrelated phrases. Labels like `NEG`, `NSCO`, `UNC`, and `USCO` were assigned based on token position and cue type.

```python
def detect_negation_uncertainty(tokens, offsets, window=5):
    ...
    if tk_lower in NEGATION_PREFIX_CUES:
        labels[i] = "NEG"
        for j in range(i+1, min(i+1+window, n)):
            if tokens[j] in SCOPE_BREAKS:
                break
            labels[j] = "NSCO"
    ...
```

Code 4. Scope-based detection of negation and uncertainty

3.3. System Optimization Statistics

We refactored the system into a class-based design with integrated preprocessing and scope detection. This version also computed cue frequencies, helping us identify the most common negation (e.g., *no*, *sense*) and uncertainty cues (e.g., *possible*, *podria*) for optimization.

3.4. Improvement of Negation Scope

In the final iteration of our rule-based system, we introduced **dependency parsing to improve the identification of scope**. Unlike the fixed window approach, this method used sentence structure to better find the parts that are negated or uncertain.

We also **expanded our lexicon of cues** to include full phrases, especially multi-word **medical expressions** in Spanish and Catalan (e.g., *no concluyente*, *sense senyals de*, *no se detecta*).

This improved model allowed us to:

- Detect expressions with pattern matching and parsing
- Capture more accurate scopes by traversing syntactic subtrees
- Reduce false positives with window-based scope detection

```python
def detect_negation_improved(text, lang="es"):
    doc = nlp_es(text) if lang == "es" else nlp_ca(text)
    negations_found = []
    for token in doc:
        if token.text.lower() in NEGATION_WORDS:
            scope = [token.text]
            for child in token.head.subtree:
                if child.dep_ not in ("punct", "cc") and child.text.lower() != token.text:
                    scope.append(child.text)
            negations_found.append(" ".join(scope))
    return list(set(negations_found))
```

Code 5. Dependency-based negation detection

3.5. Further improvements

We kept improving the approach and defined a new function analyze_medical_context that returns a better structured dictionary with the proper negated terms, uncertain terms and so on.

```python
def analyze_medical_context(text, lang="es"):
    ...
    results = {
        "negated_terms": [],
        "uncertain_terms": [],
        "double_negated_terms": [],
        "negation_cues_used": Counter(),
        "uncertainty_cues_used": Counter(),
        "medical_terms_found": Counter()
    }

    for sent in doc.sents:
        for i, token_text in enumerate(sent_tokens):
            ...

    return results
```

Code 6. Final version of the negation detection

We also added a new function, is_double_negation, which we use to determine whether a negation cue appears within the scope of another negation. This allows us to avoid incorrectly registering it as a valid cue. We use the subset of negation cues that could be used together to imply no negation.

```python
def is_double_negation(tokens):
    simple_negation_cues = {"no", "sin", "nunca", "jam s", "ning n", "ninguna", "nadie", "ninguno", "negado", "niega"}
    negation_count = sum(1 for token in tokens if token.lower() in simple_negation_cues)
    return negation_count >= 2
```

Code 7. Double negation detection function

4. Our First Conclusions and Future Improvements

Our current rule-based system works well but still has some limits in detecting the full scope of cues and covering all possible expressions. In the future, we plan to improve how the system handles multi-word cues, especially in complex cases or those specific to Catalan.

We also want to include evaluation metrics to better measure how the system performs and help us make improvements. Finally, we aim to explore hybrid models that mix rule-based methods with machine learning (and deep learning).

5. Acknowledgements

We acknowledge the articles from Slater et al. (2021) for their heuristic-based negation detection algorithm, and Argüello-González et al. (2021) for their hybrid approach to negation recognition in Spanish clinical texts. Their work inspired the development and evaluation of our rule-based system.