CS4041D Natural Language Processing

Assignment 2: Detection of Missing Information & Ambiguous Pronouns in SRS Documents

Topic Code- SW 2

REPORT (Deliverable- 2)

Group no.- 13

Prajit Sen - B220467CS
Prarthana Phukan - B220475CS
Aryan Dev Tiwari - B220194CS

6th Nov, 2025

1. Abstract

Software Requirements Specification (SRS) documents serve as the foundation for software design, development, and validation. However, real-world requirements often contain quality issues such as missing information and ambiguous pronoun usage, which reduce clarity and increase the risk of misinterpretation.

In this work, we develop an automated requirement-quality checker that analyzes SRS documents to identify two key issues:

(1) missing or placeholder information (e.g., “TBD”, “to be decided”), and (2) vague pronouns with unclear referents.

The tool uses spaCy-based linguistic parsing combined with rule-based heuristics to detect problematic sentences and automatically generate clearer rewrite suggestions. The system also produces an annotated version of the SRS text and a summary report quantifying issue occurrences. The results demonstrate that even well-structured SRS documents frequently contain ambiguous and incomplete statements, reinforcing the need for automated requirement-quality analysis tools in software engineering.

2. Objectives

The tool aims to improve the clarity of SRS documents by:

- **Detecting missing information** where placeholders such as “TBD” or “N/A” etc., indicate incomplete requirements.

- **Identifying ambiguous pronouns** whose reference is unclear or could refer to multiple nouns in the sentence.

Additionally, the tool provides **rewrite suggestions** to help make the requirements more precise and unambiguous.

3. Methodology

The Requirement Quality Checker operates in four stages: **document ingestion**, **linguistic processing**, **issue detection**, and **recommendation generation**.

3.1 Document Ingestion and Text Extraction

The system accepts multiple input formats commonly used in Software Requirement Specifications (SRS), including **PDF, DOCX, HTML, and TXT** files.
Different parsing libraries are used depending on the file type, as shown below:

File Type	Extraction Method / Library

PDF	`fitz` (PyMuPDF) for page-wise text extraction
DOCX	`mammoth` to convert Word structure into HTML and extract text
HTML	`BeautifulSoup` to extract visible HTML text content
TXT	Direct UTF-8 text reading

To maintain readability, the extracted text is preserved **line-by-line**, enabling accurate **line number reference** in the output.

3.2 Linguistic Processing

The extracted text is processed using the **spaCy NLP toolkit**, which performs:

- Tokenization
- Part-of-Speech (POS) tagging

This enables identification of:

- **Pronouns** (`it`, `they`, `this`, `these`, `those`)
- **Candidate antecedent nouns** occurring **earlier** in the sentence

The spaCy model is loaded with necessary syntactic features, while Named Entity Recognition is disabled to improve performance.

3.3 Issue Detection

3.3.1 Missing Information Detection (Type 1 Issues)

A regular expression pattern is used to detect placeholder expressions such as:

`TBD, to be decided, to be confirmed, undecided, placeholder, N/A, blank values etc.`

When such a term is found:

- The sentence is **flagged** and marked using: <<XXX>>
- A rewrite suggestion is generated using: `<specify required value>`

This ensures that requirements are written in a **complete and verifiable** manner.

3.3.2 Ambiguous Pronoun Detection (Type 2 Issues)

Ambiguous pronoun detection is based on **local antecedent evaluation**:

1. Identify pronouns from the target set.
2. Scan **previous nouns**.
3. If **two or more** valid nouns could serve as antecedents, the pronoun is considered **ambiguous**.
4. The pronoun is **marked inline** using: <<!!!>>.

This method ensures **only unclear pronouns** are flagged (not all pronouns).

3.4 System Output Formatting

The tool produces a **marked version** of the text where issues are highlighted in place:

`[Line X] <sentence with <<XXX>> or <<!!!>> inserted>`

Issue Type	Inline Marking Used	Example
Missing Information	`<<XXX>>`	`The response time is TBD <<XXX>>.`
Ambiguous Pronoun	`<<!!!>>`	`The module sends the message and stores it <<!!!>>.`
Line Indicator	`[Line X]` prefix	`[Line 12] The module stores it <<!!!>>.`

3.5 Recommendation and Rewrite Generation

Along with inline markings, a **Summary Report & Recommendations** section is generated in the following format-

For Ambiguous Pronouns:

```
Line X: <original sentence>

<<!!!>> Ambiguous Pronoun: '<pronoun>'

   - Potential Antecedents: '<noun1>', '<noun2>', ...

   @@@ Suggested Rewrites:

   - Replace '<pronoun>' with '<noun1>': "<rewritten sentence>"

   - Replace '<pronoun>' with '<noun2>': "<rewritten sentence>"
```

For Missing Information:

```
Line Y: <original incomplete sentence>

<<XXX>> Missing Information (Type 1 Issue)
```

```
@@@ Recommendation:

Replace placeholder text with a specific, measurable requirement
value.
```

3.6 Results and Issue Statistics

After processing the given SRS document, the following counts were obtained:

Metric	Meaning	Count
Missing Information (Type 1)	Incomplete requirement statements	**eg.-7**
Ambiguous Pronouns (Type 2)	Pronouns with unclear referents	**eg.-0**
Total Issues Detected	Sum of both issue types	**eg.-7**

4. Key observations and Findings

Ambiguous Pronoun issues were more frequent than Missing Information issues, indicating that the SRS is mostly complete but **some statements are unclear due to vague referent references**.

Certain placeholder terms (e.g., **“unknown”** (removed in our system)) may still be flagged in contexts where they are **used intentionally**, showing that the missing information detection is **pattern-based** and does not infer intent.

The pronoun **“that”** was removed from ambiguity detection because it commonly appears in **clear relative clause constructions**, and flagging it introduced **false positives**.

Personal and possessive pronouns (*I, we, our, your*) were excluded since SRS language should **describe system behavior**, not the perspective of authors or users.

The refined ambiguity detection now focuses on **it, they, this, these, those**, which are the pronouns that most commonly cause **uncertainty about the referenced entity**.

Some cases of pronoun clarity depend on **domain knowledge**, which **cannot be inferred through pattern matching alone**, meaning certain edge cases may still not be perfectly classified.

The system provides **automatic rewrite suggestions**, enabling direct improvement of requirement statements rather than only identifying issues.

Even with improvements, **some pronoun clarity relies on domain knowledge**, meaning rule-based detection may still miss or over-detect in edge cases.

THANK YOU