# SCIDOCA 2025 Shared Task Method Summary

## Vectorization

Vectorization is a critical step for transforming textual data into machine-readable representations, supporting tasks such as citation discovery and retrieval. Traditional techniques like TF-IDF and word embeddings (e.g., Word2Vec, GloVe) can encode fundamental semantics but often struggle with long-range contextual dependencies typical of academic texts. More advanced transformer-based models (e.g., BERT, RoBERTa) leverage self-attention to capture nuanced language patterns, yet their 512-token input limit poses challenges in processing extended passages where crucial context spans multiple sentences.

Longformer employs a hybrid attention mechanism—combining local windowed attention and sparse global attention—to process sequences of up to 4096 tokens. Unlike methods that rely on segmenting inputs into smaller chunks or sliding windows, Longformer can handle most academic paragraphs in a single pass, preserving broader contextual integrity. This capability is particularly beneficial for citation-related tasks, as it ensures that citations and their surrounding context remain intact within one continuous representation.

By adopting Longformer for vectorization, our approach maximizes context retention while keeping computational overhead manageable. This design aligns well with prior research on extended transformers, as it reinforces the importance of a more comprehensive encoding strategy for scholarly texts. Consequently, we achieve greater accuracy and reliability in extracting citation information, reflecting the growing trend toward leveraging longer sequence models for document-level understanding.

## Model

Given Text

Text cite 3

Text cite 1

CONCAT

Attention

Abstract context vector

Binary classification

Attention

Text cite 2

Candidate abstract

Candidate title

Given Text

Title context vector

The model is a binary classification system designed to predict whether a **given text** can cite a **candidate**. Each **candidate** has a **title** and an **abstract**, and each of these attributes attends to multiple **other texts**. These texts have already cited the candidate (this information is taken from the training set). As a result, the candidate generates two context vectors—one for the **title** and one for the **abstract**. These vectors are then **concatenated** with the **given text** and passed through a **fully connected layer** for final classification.

In Task 1, the given text is the paragraph, and each candidate is one of the references in the "candidate\_references" list. From the provided example:

{"paragraph": "Recent advances in natural language processing have significantly improved the performance of models on various tasks such as machine translation and question answering.",

"candidate\_references": [

"[Vaswani et al. 2017]",

"[Devlin et al. 2019]",

"[Brown et al. 2020]",

"[Radford et al. 2018]"

]}

This results in 4 samples:

1. Given text = "paragraph", Candidate = "[Vaswani et al. 2017]"
2. Given text = "paragraph", Candidate = "[Devlin et al. 2019]"
3. Given text = "paragraph", Candidate = "[Brown et al. 2020]"
4. Given text = "paragraph", Candidate = "[Radford et al. 2018]"

Given Text

Candidate title

Text cite 1

Text cite 2

Text cite 3

Candidate abstract

Attention

Attention

Title context vector

Abstract context vector

Given Text

CONCAT

Binary classification