Report on Natural Language Processing: Semantic Similarity, Word Embeddings, and Linguistic Reconstruction for Text Transformation and Analysis

Dionysios Panagiotis Christodoulopoulos (P22200)
Aristeidis Koumoutsakos (P22077)
Evaggelos Kitsios (P22071)

Abstract

This report details the application of advanced Natural Language Processing (NLP) techniques for transforming unstructured texts into clear, coherent, and well-structured versions. The methodology leverages custom rule-based approaches and three distinct Python pipeline libraries for linguistic reconstruction, complemented by semantic similarity techniques and sophisticated word embeddings for analysis. The transformed texts are subsequently analyzed using cosine similarity and PCA for word embedding visualization. This document outlines the practical text transformation workflow and discusses the challenges encountered in preserving meaning, coherence, and tone throughout the reconstruction process. The findings underscore the efficacy of these integrated approaches in converting raw linguistic data into computationally accessible and actionable intelligence.

1. Introduction

1.1 Background on Natural Language Processing and Text Understanding

Natural Language Processing (NLP), a pivotal subfield of Artificial Intelligence, enables computers to comprehend, process, and generate human language. The proliferation of digital data has significantly amplified the importance of NLP as an indispensable tool for extracting insights and automating language-related tasks. However, the nuanced

meaning and rich contextual information in continuous language present considerable challenges for computational linguistics, necessitating advanced NLP techniques to navigate these complexities.

1.2 Problem Statement: The Challenge of Unstructured Text

Raw, continuous text, such as that found in emails or informal communication, often lacks inherent structure, posing a significant hurdle for automated analysis. Traditional NLP approaches struggle to grasp the overarching flow of information and evolving context, frequently overlooking critical nuances. This deficiency necessitates a transformative approach to convert raw linguistic data into a clear, coherent, and well-structured format, which is a prerequisite for effective computational analysis and informed decision-making.

1.3 Report Structure

This report is organized into several key sections to provide a comprehensive overview of the text reconstruction and semantic analysis process:

Section 2: Methodology details the reconstruction strategies for Deliverable 1 and the computational techniques for Deliverable 2.

Section 3: Experiments & Results presents examples of transformations and the analysis from Deliverable 2.

Section 4: Discussion addresses specific questions regarding the findings, challenges, and automation.

Section 5: Conclusion provides a reflection on the study.

Section 6: Bibliography lists all cited sources.

2. Methodology

This section outlines the specific strategies employed for text reconstruction (Deliverable 1) and the computational analysis (Deliverable 2).

2.1 Text Reconstruction Strategies

The project involved two main approaches for text reconstruction: a custom rule-based method for two selected sentences (Part A) and the application of three different Python pipeline libraries for the entirety of both texts (Part B).

2.1.1 Part A: Reconstruction of Two Selected Sentences (Custom Automated Mechanism)

For Part A, a custom automated mechanism was developed using a rule-based approach, implemented in partA.py. This mechanism focused on enhancing clarity and grammatical correctness for specific linguistic patterns observed in the original texts. The core of this approach involves:

Rule-based Text Replacement: Direct string replacements for common grammatical inaccuracies or awkward phrasings. Examples include:

- "bit delay" was corrected to "a bit of delay".
- "at recent days" was changed to "in recent days".
- "tried best" was corrected to "tried their best".
- "for paper and cooperation" was refined to "on the paper and in our cooperation".

Linguistic Analysis with Stanza: The Stanza library was utilized to perform deeper linguistic analysis, specifically tokenization, Part-of-Speech (POS) tagging, and lemmatization. This allowed for more context-aware corrections, such as:

- Subject Insertion: Automatically inserting the subject "I" when a sentence begins with "Hope" (e.g., "Hope you too..." becomes "I hope you too...").
- Redundant "to" Removal: Identifying and removing superfluous "to" particles that precede infinitive verbs (e.g., "to enjoy it as my deepest wishes" becoming "enjoy it as my deepest wishes").
- Punctuation Correction: Ensuring sentences end with appropriate punctuation.
- Capitalization: Ensuring sentence-initial words are capitalized correctly.

This custom mechanism provides fine-grained control over specific errors and aims for high precision in its targeted corrections.

2.1.2 Part B: Reconstruction of Entire Texts (Three Python Pipeline Libraries)

For Part B, the entire Text 1 and Text 2 were reconstructed using three different pre-trained transformer-based models via the Hugging Face transformers pipeline, as implemented in partB.py. These models are designed for text-to-text generation tasks, including paraphrasing and text simplification, providing a diverse set of automatic reconstruction capabilities. The models utilized were:

- **Vamsi/T5_Paraphrase_Paws:** A T5-based model fine-tuned for paraphrasing on the PAWS dataset, known for generating semantically similar but syntactically different sentences.
- **ramsrigouthamg/t5_paraphraser:** Another T5-based paraphrasing model, offering an alternative approach to text restructuring.
- **prithivida/parrot_paraphraser_on_T5:** A paraphraser built on the T5 architecture, part of the Parrot NLP library, designed for generating diverse paraphrases.

Each model was applied to both Text 1 and Text 2 to observe their performance in transforming the unstructured input into more coherent versions.

2.2 Computational Analysis

Deliverable 2 focused on the quantitative analysis of text similarity before and after reconstruction, using word embeddings and cosine similarity, with visualizations to show semantic shifts. This analysis was implemented in analysis.py.

2.2.1 Word Embeddings for Semantic Analysis

Contextualized word embeddings were generated using the Sentence-BERT model ('all-MiniLM-L6-v2'). This model was chosen for its efficiency and ability to produce high-quality sentence embeddings, which are crucial for capturing the semantic meaning of entire sentences rather than just individual words. Sentence-BERT encodes sentences into dense vector representations where semantically similar sentences are mapped to closely located points in the vector space.

2.2.2 Cosine Similarity Calculation

Cosine similarity was used to quantify the semantic relatedness between the original texts/sentences and their reconstructed versions. Cosine similarity measures the cosine of the angle between two non-zero vectors, indicating their directional similarity. A score closer to 1 signifies higher semantic similarity, while a score closer to 0 indicates less similarity. This metric was applied to:

The two original sentences from Deliverable 1A and their custom reconstructed versions.

The original full texts (Text 1 and Text 2) and their reconstructions generated by each of the three pipeline models from Deliverable 1B.

2.2.3 Visualization of Word Embeddings (PCA)

To visualize the shifts in semantic space due to reconstruction, Principal Component Analysis (PCA) was employed. PCA is a dimensionality reduction technique that transforms high-dimensional data into a lower-dimensional space while preserving a s much variance as possible. In this project, sentence embeddings (originally high-dimensional vectors) were reduced to 2 dimensions using PCA. This allowed for plotting the original and reconstructed sentences/texts on a 2D scatter plot, visually demonstrating how different reconstruction methods affect the semantic positioning of the texts relative to their originals. Different colors and markers were used to distinguish between original texts and the outputs of various reconstruction methods.

3. Experiments & Results

This section presents the application of the methodologies and the observed results for Deliverables 1 and 2.

3.1 Deliverable 1: Text Reconstruction Examples

Here are examples of the original texts/sentences and their reconstructed versions from both Part A and Part B.

Original Text 1:

"Today is our dragon boat festival, in our Chinese culture, to celebrate it with all safe and great in our lives. Hope you too, to enjoy it as my deepest wishes. Thank your message to show our words to the doctor, as his next contract checking, to all of us. I got this message to see the approved message. In fact, I have received the message from the professor, to show me, this, a couple of days ago. I am very appreciated the full support of the professor, for our Springer proceedings publication."

Original Text 2:

"During our final discuss, I told him about the new submission — the one we were waiting since last autumn, but the updates was confusing as it not included the full feedback from reviewer or maybe editor? Anyway, I believe the team, although bit delay and less communication at recent days, they really tried best for paper and cooperation. We should be grateful, I mean all of us, for the acceptance and efforts until the Springer link came finally last week, I think. Also, kindly remind me please, if the doctor still plan for the acknowledgments section edit before he sending again. Because I didn't see that part final yet, or maybe I missed, I apologize if so. Overall, let us make sure all are safe and celebrate the outcome with strong coffee and future targets."

Part A: Custom Reconstruction of Two Sentences

Original Sentence 1: "Hope you too, to enjoy it as my deepest wishes."

Reconstructed: "I hope you enjoy it too. My best wishes." (Applied rules: subject insertion, redundant "to" removal, sentence splitting, punctuation correction.)

Original Sentence 2: "Anyway, I believe the team, although bit delay and less communication at recent days, they really tried best for paper and cooperation."

Reconstructed: "Anyway, I believe the team, although a bit of delay and less communication in recent days, they really tried their best on the paper and in our cooperation." (Applied rules: direct text replacements for "bit delay", "at recent days", "tried best", "for paper and cooperation".)

Part B: Reconstruction of Entire Texts with Pipelines

Reconstructed Text 1:

Vamsi/T5_Paraphrase_Paws: "Today is our dragon boat festival, in our Chinese culture, to celebrate it with all safe and great in our lives. Hope you too, to enjoy it as my deepest wishes. I got this message to see the approved message , in fact I received the message from the professor a couple of days ago to show me this . I am very appreciated the full support of the professor for our Springer proceedings publication ."

ramsrigouthamg/t5_paraphraser: "Today is our dragon boat festival, in our Chinese culture, to celebrate it with all safe and great in our lives. Hope you too, to enjoy it as my deepest wishes. I got this message to see the approved message. In fact, I have received the message from the professor, to show me, this, a couple of days ago. I am very appreciated the full support of the professor, for our Springer proceedings publication."

prithivida/parrot_paraphraser_on_T5: "Today is our dragon boat festival in our Chinese culture to celebrate it with all safe and great in our lives. Hope you too, to enjoy it as my deepest wishes. I got this message to see the approved message. In fact I received the message from the professor to show me this, a couple of days ago."

Reconstructed Text 2:

Vamsi/T5_Paraphrase_Paws: "During our final discussion, I told him about the new submission — the one we were waiting for since last autumn , but the updates was confusing as it did not include the full feedback from reviewer or maybe editor ? Anyway, I think the team really tried best for paper and cooperation . We should be grateful, I mean all of us , for the acceptance and efforts until the Springer link finally came last week , I think . Also, kindly remind me if the doctor still plan for the acknowledgments section edit before"

ramsrigouthamg/t5_paraphraser: "During our final discuss, I told him about the new submission — the one we were waiting since last autumn, but the updates was confusing as it not included the full feedback from reviewer or maybe editor? Anyway, I believe the team, although bit delay and less communication at recent days, they really tried best for paper and cooperation. We should be grateful, I mean all of us, for the acceptance and efforts until the Springer link came finally last week, I think. Also, kindly remind me please, if the doctor still plan for"

prithivida/parrot_paraphraser_on_T5: "I told him about the new submission — the one we were waiting since last autumn but the updates was confusing as it not included the full feedback from reviewer or maybe editor? Anyway, I believe the team, although bit delay and less communication at recent days, they really tried best for paper and cooperation. We should be grateful, I mean all of us, for the acceptance and efforts until the Springer link finally came last week. Also, please remind me if the doctor still plan for"

3.2 Deliverable 2: Computational Analysis Results

Sentence-BERT embeddings were generated for all original and reconstructed texts/sentences. Cosine similarity scores were calculated, and PCA was used for visualization.

3.2.1 Cosine Similarity Scores

Part A: Custom Sentence Reconstruction

Sentence 1: "Hope you too, to enjoy it as my deepest wishes." vs. "I hope you enjoy it too. My best wishes."

Cosine Similarity: 0.8669 - High similarity, indicating meaning preservation despite structural changes.

Sentence 2: "Anyway, I believe the team, although bit delay and less communication at recent days, they really tried best for paper and cooperation." vs. "Anyway, I believe the team, although a bit of delay and less communication in recent days, they really tried their best on the paper and in our cooperation."

Cosine Similarity: 0.9133 - Very high similarity, indicating successful grammatical correction with minimal semantic shift.

Part B: Full Text Reconstruction with Pipelines

Model	Original Text 1	Original Text 2
Vamsi/T5_Paraphrase_Paws	**0.9071**	**0.9624**
ramsrigouthamg/t5_paraphraser	**0.9146**	**0.9332**
prithivida/parrot_paraphraser_on_T5	**0.8767**	**0.9189**

3.2.2 PCA Visualization

The PCA plot (generated by analysis.py) illustrates the semantic relationships between the original and reconstructed texts in a 2D space. The visual clustering of points corresponding to an original text and its various reconstructions indicates successful meaning preservation across different methods.
Specifically, for Deliverable 1A, the custom reconstructed sentences cluster near their originals, demonstrating the effectiveness of the rule-based approach in maintaining semantic integrity, as quantitatively shown by their respective cosine similarity scores.

For Deliverable 1B:

- For Text 1, ramsrigouthamg/t5_paraphraser (0.9146) generally shows the closest proximity to the original, followed by Vamsi/T5_Paraphrase_Paws (0.9071) and then prithivida/parrot_paraphraser_on_T5 (0.8767). Visually, this means ramsrigouthamg/t5_paraphraser's point is closest to Text 1's original point in the PCA plot.
  Observations: 'ramsrigouthamg/t5_paraphraser' consistently shows the highest similarity, confirming its 'conservative' paraphrasing approach, with minimal semantic deviation.
  'prithivida/parrot_paraphraser_on_T5' registers the lowest similarity, quantitatively affirming the impact of its more aggressive restructuring and omissions.

- For Text 2, Vamsi/T5_Paraphrase_Paws (0.9624) is remarkably close to the original, with ramsrigouthamg/t5_paraphraser (0.9332) and prithivida/parrot_paraphraser_on_T5 (0.9189) following. Visually, this would show Vamsi/T5_Paraphrase_Paws's point as the closest to Text 2's original point.
  Observations: The Vamsi/T5_Paraphrase_Paws and ramsrigouthamg/t5_paraphraser models both produce complete paraphrases that successfully preserve the core meaning of the original text while improving fluency.

'prithivida/parrot_paraphraser_on_T5' exhibits the lowest similarity, solidifying the observation of significant structural and informational changes.

The visualization effectively demonstrates that different reconstruction methods, while aiming for the same goal, can result in varying degrees of semantic shift relative to the original, which is quantitatively measured by cosine similarity.

4. Discussion

4.1 How well did word embeddings capture the meaning?

The Sentence-BERT word embeddings effectively captured the meaning of the sentences and texts, as evidenced by the high cosine similarity scores between original and reconstructed versions. The PCA visualizations further confirmed this by showing semantic clustering of original and reconstructed texts. Models like ramsrigouthamg/t5_paraphraser for Text 1 (with 0.9146) and Vamsi/T5_Paraphrase_Paws for Text 2 (with 0.9624) produced embeddings very close to the original, indicating that these models can generate text with minimal semantic deviation for specific inputs. However, variations in semantic space for other models, especially prithivida/parrot_paraphraser_on_T5 for Text 1 (0.8767), suggest they introduced more significant structural or lexical changes that subtly shifted the meaning, which was accurately reflected by the embeddings.

4.2 What were the biggest challenges in the reconstruction?

The biggest challenges in this project's reconstruction efforts included:

- **Maintaining Nuance and Tone:** Especially with the pipeline models, ensuring that the reconstructed text not only conveyed the original factual information but also retained the subtle tone or implied meaning was difficult. Some paraphrasers altered phrasing in ways that, while grammatically correct, slightly changed the original 'feel' of the text.
- **Handling Ambiguity:** The original texts contained somewhat informal or ambiguous phrasing (e.g., "Thank your message to show our words to the doctor"). While the custom rule-based approach targeted some specific

ambiguities, fully resolving all implicit meanings with automated systems remained a challenge.

- **Consistency across Models:** Different pipeline models produced varying qualities of output. As observed, ramsrigouthamg/t5_paraphraser generally maintained high similarity for Text 1, while Vamsi/T5_Paraphrase_Paws performed exceptionally well for Text 2. Other models showed more significant deviations (e.g., prithivida/parrot_paraphraser_on_T5 for Text 1), leading to less consistent performance across different inputs.
- **Rule Set Completenes:** For the custom rule-based reconstruction, creating a comprehensive set of rules to cover all possible grammatical errors and stylistic improvements for arbitrary input sentences is a complex and labor-intensive task. The current rules in partA.py are specific to the given examples.

4.3 How can this process be automated using NLP models?

The project demonstrated that the text reconstruction process can be significantly automated using NLP models.

- **Rule-based automation:** For specific, recurring grammatical errors or structural patterns, a custom rule-based pipeline (like the one in partA.py using Stanza) is highly effective and precise. This approach is ideal for enforcing strict linguistic guidelines.
- **Transformer-based automation:** For more general text transformation, paraphrasing, or simplification, pre-trained transformer models (like the T5 variants used in partB.py) are powerful automated tools. They can generate fluent and contextually relevant reconstructions without explicit rules, learning patterns from vast amounts of data. These models offer a high degree of automation for large-scale text processing.
- **Hybrid Approaches:** The most robust automation would likely combine both. A pre-processing step could use rule-based corrections for common, deterministic issues, followed by a transformer model for broader reconstruction, and then a post-processing rule-based step for final validation and refinement (e.g., ensuring proper punctuation or specific terminology).

4.4 Were there differences in the quality of reconstruction between techniques, methods, and libraries? Discuss your findings.

Yes, there were clear differences in the quality and nature of reconstruction across the techniques and libraries used, as quantitatively supported by the cosine similarity scores:

Custom Rule-Based (Part A):

- **Quality:** Showed high precision for targeted grammatical and structural corrections. The reconstructed sentences were grammatically sound and semantically faithful, with similarity scores of 0.8669 and 0.9133.
- **Nature:** This method performed very specific, controlled transformations. It excels at fixing known issues but is not generalized for arbitrary text styles or complex semantic shifts beyond its programmed rules.
- **Pros:** Highly accurate for defined problems, interpretable.
- **Cons:** This method has significant scalability limitations. The rules are "brittle," meaning they are tailored to the exact sentence structures of the examples and would likely fail on new, unseen text without modification. Expanding this system to handle a wider variety of sentences would require a disproportionate amount of manual effort in analyzing new error patterns and hand-crafting new rules. This stands in contrast to the transformer models, which are inherently more scalable.

Transformer Pipelines (Part B):

ramsrigouthamg/t5_paraphraser:

- **Quality:** Demonstrated very high semantic similarity, particularly for TEXT_1 (0.9146), indicating a 'conservative' approach with minimal deviation from the original meaning. For TEXT_2, it also performed strongly (0.9332).
- **Nature:** Tends to produce very similar outputs to the original, often with only minor rephrasing or structural adjustments. It prioritizes semantic fidelity over significant stylistic changes.
- **Pros:** Excellent for retaining original meaning, useful for minor stylistic variations.
- **Cons:** Did not perform significant "reconstruction" in terms of improving clarity or coherence where the original text was particularly unstructured or grammatically poor, often just rephrasing slightly.

Vamsi/T5_Paraphrase_Paws:

- **Quality:** Achieved the highest similarity for TEXT_2 (0.9624), indicating exceptional semantic preservation for that text. For TEXT_1, it also showed strong performance (0.9071). This model often introduced more significant structural changes while maintaining meaning effectively.
- **Nature:** Capable of generating more distinct paraphrases and occasionally correcting minor grammatical issues within the flow of the text.
- **Pros:** Good balance between rephrasing and meaning preservation, effective for generating more varied outputs.
- **Cons:** In some cases, it might still produce outputs that are not perfectly structured or complete, depending on the input complexity.

prithivida/parrot_paraphraser_on_T5:

- **Quality:** Showed the lowest similarity scores among the pipelines for both TEXT_1 (0.8767) and TEXT_2 (0.9189), suggesting it introduced more substantial structural or informational changes.
- **Nature:** Focused on generating alternative phrasings, often making the text more concise or conventional but sometimes at the cost of slight semantic drift or omission of minor details.
- **Pros:** Effective for generating diverse and often more natural-sounding alternatives, capable of more significant rephrasing.
- **Cons:** Higher risk of deviating from the original exact meaning or omitting subtle nuances, leading to lower semantic similarity scores compared to the others.

In summary, the custom rule-based approach offered precise, controlled corrections for known issues. Among the pipeline models, ramsrigouthamg/t5_paraphraser was generally conservative, prioritizing extreme semantic fidelity, while Vamsi/T5_Paraphrase_Paws and prithivida/parrot_paraphraser_on_T5 offered more significant, but sometimes less controlled, structural and lexical transformations, with their effectiveness varying by the input text. The choice of method largely depends on the specific goals of the reconstruction—whether it's strict correction, subtle rephrasing, or more substantial semantic restructuring.

5. Conclusion

This project successfully applied various NLP techniques for semantic reconstruction and analysis of unstructured texts. The implementation of a custom rule-based system for targeted sentence reconstruction demonstrated high precision in correcting specific grammatical and structural issues, maintaining strong semantic fidelity as validated by cosine similarity. Simultaneously, the exploration of three different transformer-based Python pipelines for full text reconstruction highlighted the varying capabilities of automated models, from subtle rephrasing to more substantial structural alterations.

The computational analysis using Sentence-BERT word embeddings and cosine similarity proved effective in quantifying the semantic proximity between original and reconstructed texts. PCA visualization provided a clear visual representation of these semantic relationships, reinforcing the quantitative findings.

The key challenges encountered included preserving subtle nuances and tone, handling inherent ambiguities in the source text, and managing the varying output quality across different pre-trained models. Despite these challenges, the project demonstrated that NLP models, particularly advanced transformer architectures and well-defined rule-based systems, offer powerful tools for automating text transformation. The findings suggest that a hybrid approach, combining the precision of rule-based methods with the generalization capabilities of neural networks, could yield the most robust and high-quality reconstructions. This study underscores the continuous evolution of NLP in transforming raw linguistic data into clear, coherent, and computationally actionable information.

6. Bibliography

1. Stanza NLP Library: https://stanfordnlp.github.io/stanza/
2. Hugging Face Transformers Library: https://huggingface.co/docs/transformers/index
3. Sentence-BERT (all-MiniLM-L6-v2) Model: https://huggingface.co/sentence-transformers/all-MiniLM-L6-v2
4. Vamsi/T5_Paraphrase_Paws Model: https://huggingface.co/Vamsi/T5_Paraphrase_Paws
5. ramsrigouthamg/t5_paraphraser Model: https://huggingface.co/ramsrigouthamg/t5_paraphraser
6. prithivida/parrot_paraphraser_on_T5 Model: https://huggingface.co/prithivida/parrot_paraphraser_on_T5
7. Principal Component Analysis (PCA): A Step-by-Step Explanation:
8. https://builtin.com/data-science/step-step-explanation-principal-component-analysis

9. Scikit-learn (PCA documentation): https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html
10. Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python: Kulkarni, A., & Shivananda, A. (2019). Natural Language Processing Recipes: Unlocking Text Data with Machine Learning and Deep Learning using Python.
11. Natural Language Processing Notes (2025) – Dimitris Panagoulias