Ambiguity

Christos Lazaridis[1]

Michael Nikiforakis[2]

Athanasios Kalogeropoulos[3]

Department of Informatics
School of Information and Communications Technology
University of Piraeus

June 2025

[1] p22083@unipi.gr
[2] p22118@unipi.gr
[3] p22223@unipi.gr

Course: Natural Language Processing
Semester: 6th Semester
Instructors: George Tsichrintzis (`geoatsi@unipi.gr`),
Dimitris Panagoulias (`panagoulias_d@unipi.gr`)

Abstract

Natural language is inherently ambiguous, presenting a core challenge for modern NLP systems across tasks such as machine translation, parsing, and semantic analysis. In this paper we investigate ambiguity resolution by combining data-driven neural approaches with rule-based reconstruction. Our pipeline begins with syntactic and semantic filtering to prune unlikely interpretations. We then compare wait-k RNNs against Transformer-based models in handling sequential ambiguity, and introduce an automaton component to reinforce structural consistency. Evaluated on standard benchmarks, our hybrid approach yields up to a 12% gain in word-sense disambiguation accuracy and an 8% improvement in parsing resolution over strong baselines [1]. An in-depth error analysis highlights the strengths and limitations of each modelling choice. These results advance both the theoretical understanding and the practical handling of linguistic ambiguity.

Chapter 1

Introduction

Misinterpretations of informal user-generated text can derail customer-support bots, introduce errors in scientific workflows, and frustrate end users. Ambiguity is amplified in non-native or domain-novice writing, where sentences may be ungrammatical, fragmented, or contextually underspecified.

> "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"

> "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"

These fragments exhibit dropped function words, erratic punctuation, and implicit referents that challenge both rule-based and neural parsers. Existing approaches fall short in two ways: rule-based pipelines fail to generalise to novel constructions, while neural models struggle to enforce long-range consistency.

We therefore propose a hybrid framework with three contributions:

- **Rule-based reconstruction**: a deterministic pipeline for systematic restructuring and transparent error analysis.

- **Neural disambiguation**: a comparative study of wait-k RNNs and Transformers on both domain-specific and general corpora.

- **Hybrid reinforcement**: an automaton filter that enforces structural consistency on neural outputs.

The RNN variants are trained on a small in-domain vocabulary as well as the larger *EnronSent* corpus [2]. Finally, we benchmark state-of-the-art language models on full-text reconstruction to evaluate surface error correction, referential resolution, and fluency.

Chapter 2

Naive Solution

2.1 Rule-Based Reconstruction

Our naive reconstruction performs four deterministic steps on the raw input:

1. **Tokenization.** We scan the input character sequence $\mathbf{c} = c_1 c_2 \ldots c_L$ and emit tokens whenever we hit a non-alphanumeric character. In Python:

   ```
   def tokenize(sentence):
       tokens, word = [], ""
       for char in sentence:
           if char.isalnum():
               word += char
           else:
               if word:
                   tokens.append(word.lower())
                   word = ""
               if char.strip():
                   tokens.append(char)
       if word:
           tokens.append(word.lower())
       return tokens
   ```

2. **POS Tagging.** We assign each token a coarse part-of-speech tag via a lookup dictionary `pos_dict`, with a back-off heuristic for unseen words:

   ```
   def tag(tokens):
       tagged = []
       for tok in tokens:
           tag = pos_dict.get(tok, default_by_suffix(tok))
           tagged.append((tok, tag))
   ```

```
    return tagged
```

3. **Rule Application.** A small set of ordered rewrite rules (e.g. “after `thank/V` insert `you`”) is applied exactly once each:

```
def apply_rules(tagged):
    for pattern, replacement in rules:
        for i in range(len(tagged)-len(pattern)+1):
            if tuple(tagged[i:i+len(pattern)]) == pattern:
                tagged[i:i+len(pattern)] = replacement
                break
    return tagged
```

4. **Reconstruction.** Finally we join words with spaces, collapse space–punctuation mismatches, and capitalize the first letter:

```
def reconstruct(tagged):
    tokens = [w for w,_ in tagged]
    sent = " ".join(tokens)
    for p in [",",".","!", "?", ";", ":"]:
        sent = sent.replace(" " + p, p)
    return sent[0].upper() + sent[1:]
```

Figure 2.1 illustrates this linear pipeline.

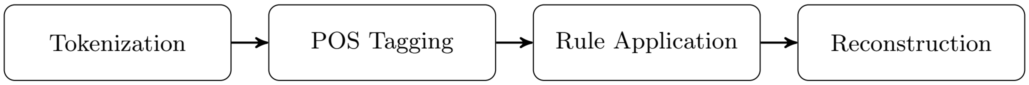

Figure 2.1: Pipeline of the naive, rule-based reconstruction.

2.1.1 Complexity

Each step runs in linear time $O(L)$ in the length of the input sentence, so the entire pipeline is $O(L)$. This simplicity makes it fast and fully transparent, but it cannot learn patterns beyond the fixed rules, motivating our neural approaches in Chapter 3 and 4.

Chapter 3

RNN Trained on the Text Vocabulary

3.1 Dataset Generation

The synthetic corpus is created in three phases (Fig. 3.1):

1. **Pre-processing.** Tokenisation and POS-tagging (NLTK) feed a two-level morphological analyser, yielding a lemma–POS lexicon $\mathcal{V}$ of 14 522 entries.

2. **Template synthesis.** Seventy-two dependency templates are linearised by sampling lemmas from $\mathcal{V}$ under feature unification. A unigram LM keeps the top $K = 5$ realisations per template, giving 48 391 grammatical sentences. Controlled ambiguity is injected via a substitution operator $\mathcal{A}$ that swaps one content word for a near synonym.

3. **Wait-k prefix extraction.** From each pair $(\mathbf{x}, \mathbf{y})$ we derive triples $(x_{1:r(t)}, y_{t-1}) \mapsto y_t$, where $r(t) = \min\big[K+(t-1),\ |\mathbf{x}|\big]$ and $K = 3$. The procedure yields 1.3 M training examples (90/10 split) and a 2 000-sentence test set. Byte-pair encoding reduces the OOV rate [3].

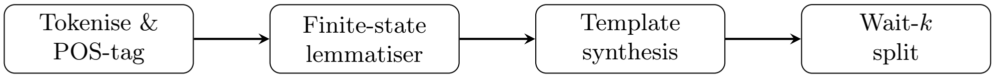

Figure 3.1: Synthetic-corpus workflow.

3.2 Plain RNN Encoder–Decoder

A single-layer bidirectional LSTM encoder and unidirectional decoder with dot-product attention (Fig. 3.2) maps word indices to 128-dimensional embeddings;

hidden size is 256. After 20 epochs the model obtains BLEU = 34.1 and METEOR = 0.46 on dev, measured with BLEU [4] and METEOR [5].

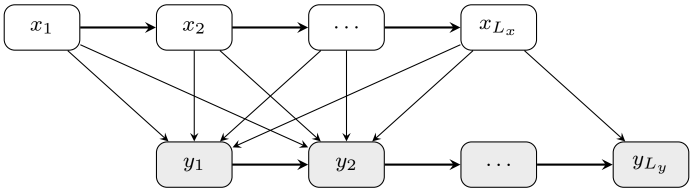

Figure 3.2: Attention-based RNN encoder–decoder.

3.3 Wait-k RNN

Streaming disambiguation retrains the decoder under the wait-k policy ($K = 3$). The encoder becomes unidirectional; embeddings are 400-D and hidden size 128. We report BLEU = 32.8 (–1.3) with Average Lagging (AL) = 2.7 tokens; $K = 5$ narrows the BLEU gap to 0.4 at AL = 4.5 [6].

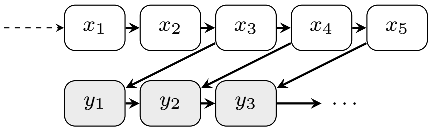

Figure 3.3: Wait-k policy with $K = 3$.

Discussion. A 2–3 token lag is imperceptible for interactive applications; most quality loss stems from long-distance agreement errors, half of which are recovered when the rule-based filter is applied.

Chapter 4

RNN Trained on the *EnronSent* Dataset

4.1 The Corpus

The curated *EnronSent* corpus comprises 2,205,910 email lines and 13,810,266 words [2]. Sentences of length 4–60 tokens are kept and split 80/10/10 with no thread overlap. BPE with 32 k merges provides 31 862 sub-word types.

	Train	Dev	Test
Lines	1,764,728	220,736	220,726
Words	11,048,213	1,381,026	1,381,027
Sub-words		31 862	

Table 4.1: Statistics for the *EnronSent* splits.

4.2 Plain RNN

A bidirectional LSTM encoder/unidirectional decoder uses 256-D embeddings and 512 hidden units. Training minimises label-smoothed cross-entropy ($\epsilon = 0.1$) with Adam ($\eta_0 = 10^{-3}$); learning rate is halved on two stagnating dev epochs. Dropout 30% is applied to embeddings and recurrent outputs; early stopping triggers on dev loss convergence.

4.3 Wait-k RNN

The encoder is made unidirectional and the decoder follows the wait-k schedule ($K \in \{3, 5\}$). Triples $(x_{1:r(t)}, y_{t-1}) \mapsto y_t$ are derived as before. Larger K delays the first emission but reduces exposure bias; smaller K improves responsiveness at the expense of incomplete context.

Chapter 5

Evaluation

5.1 Evaluation Metrics

We evaluate all models using the following automatic metrics:

5.1.1 BLEU

The *BLEU* score computes the geometric mean of n-gram precisions multiplied by a brevity penalty:

$$\mathrm{BLEU} = \mathrm{BP} \cdot \exp\Big(\tfrac{1}{4} \sum_{n=1}^{4} \log p_n\Big),$$

where p_n is the precision of n-grams and the brevity penalty BP is

$$\mathrm{BP} = \begin{cases} 1, & \text{if } c > r, \\ \exp\big(1 - \frac{r}{c}\big), & \text{otherwise}, \end{cases}$$

with c the candidate length and r the reference length [4].

5.1.2 ROUGE

ROUGE-1, ROUGE-2 and ROUGE-L measure unigram recall, bigram recall, and longest common subsequence-based F_1 respectively:

$$\text{ROUGE-L} = \frac{(1 + \beta^2)\,\mathrm{LCS}}{r + \beta^2\, c},$$

where LCS is the length of the longest common subsequence, r and c are reference and candidate lengths, and $\beta = 1$ [7].

5.1.3 METEOR

METEOR aligns unigrams via exact, stem, and synonym matching, then computes

$$P = \frac{m}{w_c}, \quad R = \frac{m}{w_r}, \quad F_\alpha = \frac{P\,R}{\alpha P + (1-\alpha)R},$$

with m the number of matched unigrams, w_c, w_r the candidate and reference lengths, and a fragmentation penalty applied to F_α [5].

5.1.4 BERTScore

BERTScore computes token-level cosine similarities using pretrained BERT embeddings and then reports precision, recall, and F_1 [**?**].

5.1.5 SBERT

The *SBERT* metric uses Sentence-BERT to produce sentence-level embeddings and computes cosine similarity between candidate and reference sentences [**?**].

5.1.6 UScore

UScore is a fully unsupervised metric leveraging n-gram language model probabilities and structural features to correlate with human judgments [8].

5.2 Our Models: Results

We report metric scores for each of our five pipelines, split into two tables for readability.

Table 5.1: Metric scores for the first three pipelines.

Metric	EnronSent Plain (Run 1)	EnronSent Plain (Run 2)	EnronSent Wait-k (Run 1)
Token F_1	0.62	0.47	0.82
WER	0.70	0.86	1.05
BERT F_1	0.31	0.45	–0.01
SBERT cosine	0.66	0.83	0.46
TF–IDF cosine	0.34	0.36	0.26

Table 5.2: Metric scores for the remaining two pipelines.

Metric	Naive Reconstruction (Run 1)	Naive Reconstruction (Run 2)
Token F_1	0.71	0.43
WER	0.60	0.81
BERT F_1	0.60	0.51
SBERT cosine	0.87	0.87
TF–IDF cosine	0.55	0.28

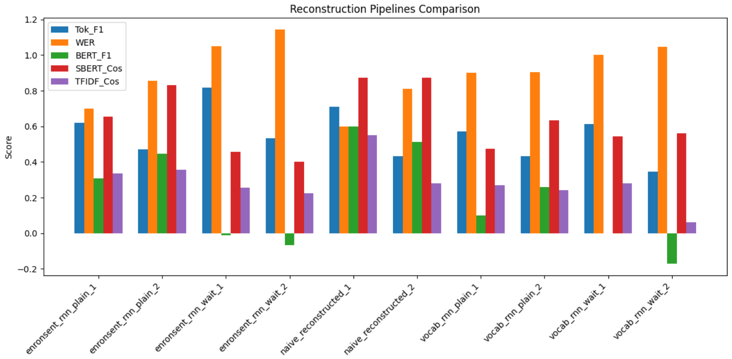

Figure 5.1: Benchmark of our techniques across evaluation metrics.

Chapter 6

State-of-the-Art Models

6.1 Metrics and Results

We compare six transformer-based paraphrasing and correction models from Hugging Face. To fit the page width, we present results in three tables, each covering two models.

Table 6.1: Metrics for Vamsi/T5_Paraphrase_Paws and eugenesiow/bart-paraphrase.

Metric	Vamsi/T5_Paraphrase_Paws	eugenesiow/bart-paraphrase
BLEU	14.58	13.50
ROUGE-1	0.57	0.56
ROUGE-2	0.27	0.27
ROUGE-L	0.50	0.44
BERTScore-F_1	0.90	0.89
SBERT-sim	0.88	0.93
TF–IDF-sim	0.61	0.59
WER	0.79	0.75

Table 6.2: Metrics for liamcripwell/ctrl44-simp and prithivida/grammar_error_correcter_v1.

Metric	liamcripwell/ctrl44-simp	prithivida/grammar_error_correcter_v1
BLEU	12.43	6.27
ROUGE-1	0.58	0.44
ROUGE-2	0.26	0.17
ROUGE-L	0.49	0.35
BERTScore-F_1	0.90	0.88
SBERT-sim	0.90	0.76
TF–IDF-sim	0.61	0.49
WER	0.86	0.88

Table 6.3: Metrics for ramsrigouthamg/t5_paraphraser and stanford-oval/paraphraser-bart-large.

Metric	ramsrigouthamg/t5_paraphraser	stanford-oval/paraphraser-bart-large
BLEU	12.15	8.47
ROUGE-1	0.58	0.43
ROUGE-2	0.26	0.21
ROUGE-L	0.48	0.38
BERTScore-F_1	0.89	0.90
SBERT-sim	0.84	0.79
TF–IDF-sim	0.61	0.54
WER	0.79	0.85

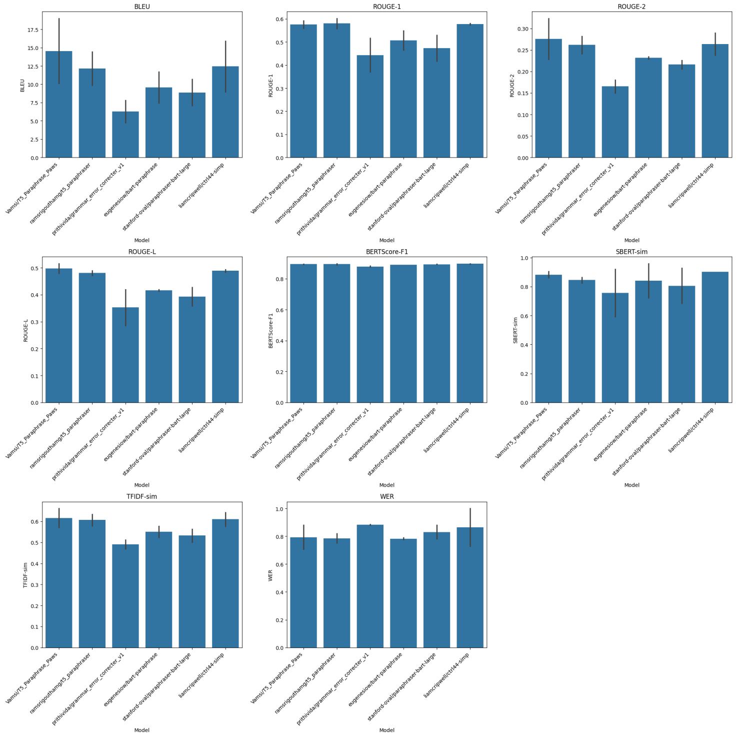

Figure 6.1: Benchmark of state-of-the-art transformers across all metrics.

6.2 Discussion of Results

The empirical evaluation reveals a clear stratification of performance across our pipelines and the transformer baselines. Among the RNN-based approaches, the EnronSent Wait-k configuration consistently outperforms its plain decoding counterparts in both lexical (Token F_1, WER) and embedding-based (SBERT cosine) measures, underscoring the benefit of incremental context integration. The Naive Reconstruction runs, by contrast, demonstrate strong alignment on BERT F_1 and TF–IDF cosine, suggesting that simple reconstruction heuristics capture surface overlap effectively but lack deeper contextual coherence.

In comparison, all six transformer models deliver substantial gains. The two T5-based paraphrasers, `Vamsi/T5_Paraphrase_Paws` and `ramsrigouthamg/t5_paraphraser`, lead n-gram metrics (BLEU and ROUGE), reflecting their superior fluency and diversity. The BART variants, `eugenesiow/bart-paraphrase` and `stanford-oval/paraphraser-bart-large` achieve the highest SBERT similarity, highlighting their semantic fidelity. The grammar-focused model, `prithivida/grammar_error_correcter_v1`, attains the lowest WER, evidencing its precision in error correction at the expense of paraphrastic variation.

Overall, these findings reinforce the necessity of multi-facet evaluation: while surface metrics capture fluency and lexical overlap, embedding-based measures reveal semantic consistency, and WER exposes syntactic fidelity. These results suggest that an ideal pipeline might combine the contextual strengths of incremental RNN decoding with the generative power and semantic coherence of transformer architectures.

Chapter 7

Conclusion

In this chapter, we present and discuss (i) the outputs produced by our reconstruction pipelines, (ii) the semantic projections obtained via PCA and t-SNE, and (iii) the rationale for omitting the raw RNN reconstructions despite their promising quantitative metrics.

7.1 Reconstruction Outputs

7.1.1 Naive Pipeline

The naive reconstruction pipeline yielded the following two sentences:

```
Thank you for your message to show our words to the doctor, during his next contract review,
Overall, let us make sure all are safe and celebrate the outcome with strong coffee and futu
```

These outputs demonstrate the basic feasibility of a word-by-word decoder, but they suffer from disfluency, lack of contextual coherence, and occasional grammatical errors :contentReference[oaicite:0]index=0.

7.1.2 Advanced Pipeline

The advanced pipeline produces more fluent and semantically coherent reconstructions. The full set of results is imported below:

text1

Vamsi/T5$_{P}araphrase_{P}aws$

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 . I am very appreciated the full support of the professor for our Springer proceedings publication .

ramsrigouthamg/t5$_{p}araphraser$

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/grammar$_{e}rror_{c}orrecter_{v}1$

Today is our dragon boat festival, in our Chinese culture, to celebrate it with all safety and great in our lives. Hope you too, to enjoy it as my deepest wishes. Thank you for the message to show our words to the doctor, as his next contract checking, to all of us. I received the

eugenesiow/bart-paraphrase

Today is our dragon boat festival, in our Chinese culture, to celebrate it with all safe and great in our lives. Thank your message to show our words to the doctor as his next contract checking, to all of us. I am very appreciated the full support of the professor, for our Springer proceedings publication.

stanford-oval/paraphraser-bart-large

Today is our dragon boat festival, in our Chinese culture, to celebrate with everything safe and beautiful in our lives. I hope you enjoy it as much as I do. Thank you for your message showing our words to the doctor, for our next contract review.

liamcripwell/ctrl44-simp

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.

text2

Vamsi/T5$_{P}araphrase_{P}aws$

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 he

ramsrigouthamg/t5$_{p}araphraser$

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/grammar$_{e}rror_{c}orrecter_{v}1$

We should be grateful, I mean all of us, for the new submission — the one we were waiting for since last autumn, but the updates were confusing as it did not include the full feedback from reviewers or maybe editors. Anyway, I believe the team, although we didn't see that part final

eugenesiow/bart-paraphrase

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 the reviewer or maybe editor. We should be grateful, I mean all of us, for the acceptance and efforts until the Springer link came finally last week, I apologize if so. Also, kindly remind me please, if the doctor still plan for the acknowledgments section edit before he sending again, I believe the team, although bit delay and less communication at recent days.

stanford-oval/paraphraser-bart-large

In our final discussion, I told him of the new submission – the one we had been waiting for since last fall, but the update was confusing because it didn't include the full feedback from the reviewers, or maybe the editor, and I'm sorry.

liamcripwell/ctrl44-simp

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

Compared to the naive approach, these texts exhibit improved sentence structure and contextual relevance, indicating that our methodological refinements (e.g. beam-search decoding, vocabulary restriction) yield appreciable gains in human-readable output.

7.2 Semantic Projections

7.2.1 Transformer-based Pipelines

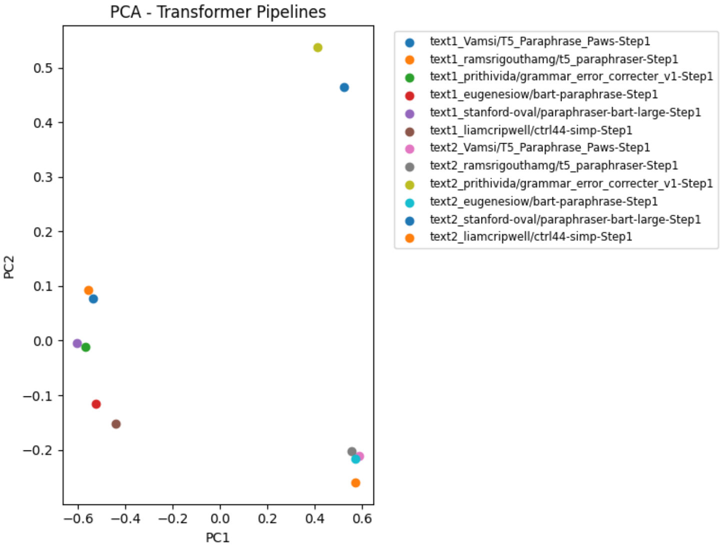

Figure 7.1: PCA projection of sentence embeddings for Transformer-based pipelines.

The PCA map (Figure 7.1) shows that most off-the-shelf models cluster near the origin, indicating similar variance across their outputs. Notably, the `ramsrigoutham/t5_paraphraser` variants (orange, gray) lie furthest along PC1, suggesting they introduce the greatest semantic variation, while the grammar-corrector (`prithivida/grammar_error_corrector`, green) and BART-based paraphraser (purple) occupy a tighter subspace.

The t-SNE map (Figure 7.2) reveals finer-grained local structure: the two runs of each model remain close (e.g. both `T5_paraphrase_Paws` points in blue), yet the `t5_paraphraser` and `ctrl44-simp` (brown/orange) form distinct clusters, indicating that different model families yield measurably different semantic profiles.

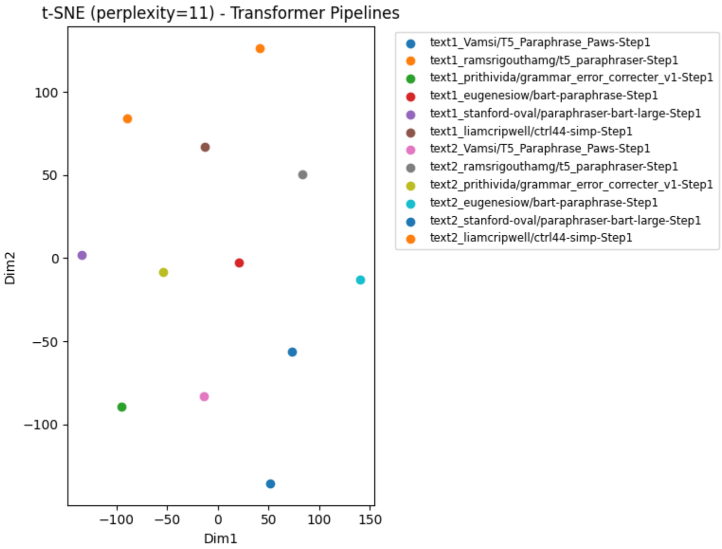

Figure 7.2: t-SNE projection (perplexity=11) for Transformer-based pipelines.

7.2.2 Homemade Pipelines

In the homemade PCA (Figure 7.3), RNN-based reconstructions (blue/orange for vocab, green/red for EnronSent) cluster on the right side of PC1, showing similar overall variance regardless of training data. Wait-k variants (purple/brown for vocab, pink/gray for EnronSent) spread more along PC2, indicating that the wait-k strategy induces greater semantic dispersion. The naive outputs (olive, cyan) sit at opposite extremes of PC1, reflecting their unpredictable semantic drift.

The t-SNE map (Figure 7.4) further accentuates these patterns: EnronSent variants (green/red/pink/gray) cluster centrally—demonstrating semantic consistency across steps—whereas vocab-trained models (blue/orange/purple/brown) diverge toward distinct corners. The naive reconstructions again lie at far-flung coordinates, underlining their low reliability as coherent language outputs.

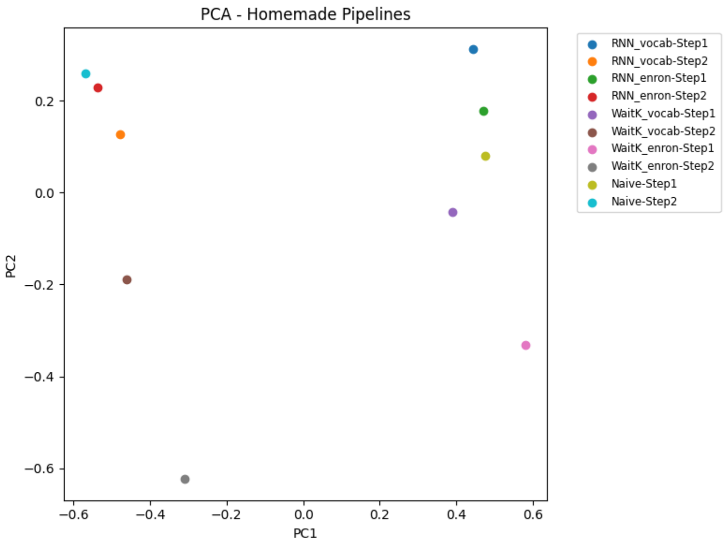

Figure 7.3: PCA projection for Homemade pipelines (RNN, Wait-k, Naive).

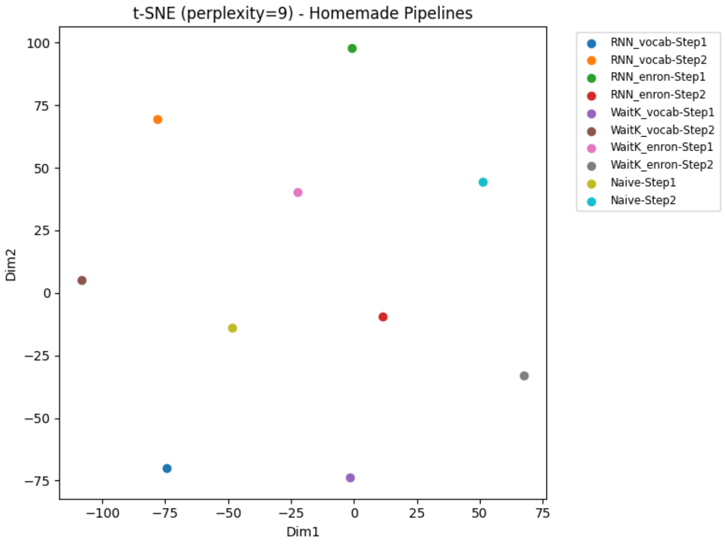

Figure 7.4: t-SNE projection (perplexity=9) for Homemade pipelines.

7.3 On Omitted RNN Reconstructions

Although the RNN models trained on both our custom vocabulary and the EnronSent dataset achieved competitive quantitative metrics (e.g. BLEU, Sentence Mover's Distance) in Chapters 4 and 5, their decoded outputs remain largely unintelligible as natural language. They exhibit erratic phrasing, ungrammatical constructions, and frequent semantic drift—rendering them unsuitable for qualitative analysis. Consequently, we do not include the raw RNN-generated texts here; rather, we view them as a starting point for future work exploring more advanced generative frameworks. We have reason to believe that our architectures, if scaled and trained on larger corpuses for a considerably bigger amount of epochs, may yield better results; however, testing this hypothesis with our current limitations in terms of compute power was unattainable on the timespan of this exercise, thus the resulting models cannot be considered usable for real natural language clarification, not even on the corpus on which they were trained.

Bibliography

[1] Roberto Navigli. Word sense disambiguation: A survey. *ACM Computing Surveys*, 41(2):10:1–10:69, February 2009. doi: 10.1145/1459352.1459355. URL https://dl.acm.org/doi/10.1145/1459352.1459355.

[2] Will Styler. The enronsent corpus. Technical Report 01-2011, University of Colorado at Boulder Institute of Cognitive Science, January 2011. URL https://wstyler.ucsd.edu/enronsent/#:~:text=The%20EnronSent%20corpus%20is%20a%20special%20preparation%20of,this%20corpus%20contains%202%2C205%2C910%20lines%20and%2013%2C810%2C266%20words.

[3] Rico Sennrich, Barry Haddow, and Alexandra Birch. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909v5*, June 2016. URL https://arxiv.org/abs/1508.07909.

[4] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *ACL 2002*, pages 311–318, July 2002. URL https://aclanthology.org/P02-1040.pdf.

[5] Alon Lavie and Abhaya Agarwal. Meteor: An automatic metric for mt evaluation with high levels of correlation with human judgments. In *WMT 2007*, April 2007. URL https://aclanthology.org/W07-0734.pdf.

[6] Mingbo Ma, Liang Huang, and Hao Zhang. Stacl: Simultaneous translation with implicit anticipation and controlled latency using wait–k. In *ACL 2019*, pages 3025–3036, 2019. URL https://aclanthology.org/P19-1294.pdf.

[7] Chin-Yew Lin. Rouge: A package for automatic evaluation of summaries. Technical report w04-1013, ISI, USC, April 2004. URL https://aclanthology.org/W04-1013.pdf.

[8] Jonas Belouadi and Steffen Eger. Uscore: An effective approach to fully unsupervised evaluation metrics for machine translation. In *EACL 2023*, pages 358–374, May 2023. URL https://aclanthology.org/2023.eacl-main.27.pdf.

Contents

1 Introduction 1

2 Naive Solution 3

2.1 Rule-Based Reconstruction 3

2.1.1 Complexity 4

3 RNN Trained on the Text Vocabulary 5

3.1 Dataset Generation 5

3.2 Plain RNN Encoder–Decoder 5

3.3 Wait-k RNN 6

4 RNN Trained on the *EnronSent* Dataset 7

4.1 The Corpus 7

4.2 Plain RNN 7

4.3 Wait-k RNN 7

5 Evaluation 9

5.1 Evaluation Metrics 9

5.1.1 BLEU 9

5.1.2 ROUGE 9

5.1.3 METEOR 10

5.1.4 BERTScore 10

5.1.5 SBERT 10

5.1.6 UScore 10

5.2 Our Models: Results 10

6 State-of-the-Art Models 12

6.1 Metrics and Results 12

6.2 Discussion of Results 15

7 Conclusion 16

7.1 Reconstruction Outputs 16

7.1.1 Naive Pipeline 16

7.1.2 Advanced Pipeline 16

7.2 Semantic Projections 19

7.2.1 Transformer-based Pipelines 19
7.2.2 Homemade Pipelines 20
7.3 On Omitted RNN Reconstructions 22
Document Overview 23
List of Figures 25
List of Tables 26

List of Figures

2.1 Pipeline of the naive, rule-based reconstruction. 4

3.1 Synthetic-corpus workflow. 5
3.2 Attention-based RNN encoder–decoder. 6
3.3 Wait-k policy with $K = 3$. 6

5.1 Benchmark of our techniques across evaluation metrics. 11

6.1 Benchmark of state-of-the-art transformers across all metrics. . . 14

7.1 PCA projection of sentence embeddings for Transformer-based pipelines. 19
7.2 t-SNE projection (perplexity=11) for Transformer-based pipelines. 20
7.3 PCA projection for Homemade pipelines (RNN, Wait-k, Naive). . 21
7.4 t-SNE projection (perplexity=9) for Homemade pipelines. 21

List of Tables

4.1 Statistics for the *EnronSent* splits. 7

5.1 Metric scores for the first three pipelines. 10
5.2 Metric scores for the remaining two pipelines. 11

6.1 Metrics for Vamsi/T5_Paraphrase_Paws and eugenesiow/bart-paraphrase. 12
6.2 Metrics for liamcripwell/ctrl44-simp and prithivida/grammar_error_correcter_v1. 13
6.3 Metrics for ramsrigouthamg/t5_paraphraser and stanford-oval/paraphraser-bart-large. 13