Traduzione Automatica IT EN con Seq2Seq Transformer

October 11, 2025

Contesto del Progetto

- La **Neural Machine Translation** (NMT) è un pilastro del Deep Learning in NLP, con ampie applicazioni industriali.
- Evoluzione storica: dai modelli sequenziali (RNN, LSTM) $\rightarrow$ dominanza attuale del **Transformer** (Vaswani et al., 2017).
- Utilizzi concreti: traduzione di documentazione tecnica, descrizioni e-commerce, interfacce multilingua per assistenti AI, comunicazione globale.
- Obiettivo del progetto: sviluppare da zero un sistema di traduzione unidirezionale **Inglese** $\leftrightarrow$ **Italiano**, addestrato integralmente sui dati forniti.

Obiettivi Principali (1/2)

Implementare e addestrare un sistema NMT completo:

- Architettura **Transformer Encoder–Decoder** ottimizzata per sequenze lunghe.
- Addestramento su CPU con batch size 32 per 9 epoche.
- Dataset di $\sim$1.9M frasi parallele EN–IT.
- Tokenizzazione custom con vocabolari dedicati e token speciali.
- Salvataggio checkpoint a ogni epoca + logging dettagliato del training.

Obiettivi Principali (2/2)

Analizzare e validare le prestazioni:

- **Inferenza** con approcci multipli:
  - Greedy decoding (baseline veloce)
  - Beam search (ottimizzazione della probabilità globale)
- **Metriche quantitative**:
  - BLEU score, token accuracy, log-likelihood
- **Analisi qualitativa** per individuare punti di forza e criticità della traduzione.
- **Selezione del miglior modello** anche in assenza di ground truth esplicita.

Struttura della Presentazione

1. Introduzione e obiettivi
2. Background teorico: Transformer e NMT
3. Dataset e preprocessamento
4. Architettura e dettagli implementativi
5. Inferenza: greedy e beam search
6. Analisi quantitativa e qualitativa
7. Visualizzazione dell'addestramento
8. Scelta del modello finale
9. Conclusioni e sviluppi futuri

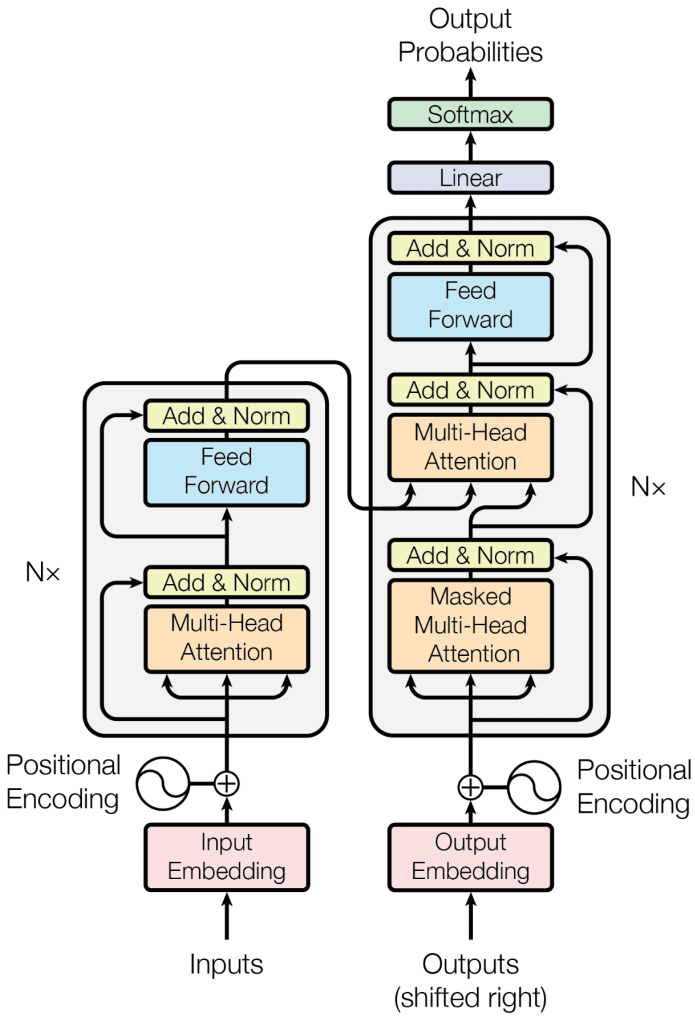
Background Teorico – Introduzione

- La **Machine Translation** è un tipico problema Seq2Seq: mappare una sequenza in ingresso (es. IT) in una sequenza in uscita (es. EN).
- La qualità della traduzione dipende da:
  - Comprensione semantica complessiva
  - Preservazione della struttura sintattica
  - Coerenza lessicale e stilistica
- Oggi i modelli **Transformer** rappresentano lo *standard de facto* per questo compito.

Dalle RNN/LSTM al Transformer

- Prima del 2017: traduzione basata su RNN e LSTM con attention.
- Limiti principali:
  - Computazione sequenziale $\rightarrow$ difficile parallelizzare
  - Prestazioni scarse su frasi lunghe
- Transformer (Vaswani et al., 2017):
  - Basato esclusivamente su attention
  - Computazione completamente parallela
  - Migliori prestazioni e scalabilità

Transformer: Architettura Encoder–Decoder



Struttura base: encoder per rappresentare la frase sorgente, decoder per generare la traduzione target (Vaswani et al., 2017).

Encoder e Decoder

Encoder:

- Stack di N layer identici
- Ogni layer = self-attention + feed-forward
- Trasforma l'input in rappresentazioni vettoriali ricche

Decoder:

- Stack di N layer
- Mascheramento per generazione autoregressiva
- Cross-attention sulle rappresentazioni dell'encoder
- Genera un token per volta fino a <eos>

Multi-Head Attention

- Ogni “testa” elabora una proiezione diversa di Q, K, V.
- Permette al modello di cogliere simultaneamente relazioni sintattiche e semantiche diverse.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_h)W^O$$

Positional Encoding

- Il Transformer non conosce l'ordine dei token: bisogna fornirglielo.
- Positional encoding = pattern sinusoidali sommati agli embedding.
- Formula:

$$PE_{(pos,2i)} = \sin\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right), \quad PE_{(pos,2i+1)} = \cos\left(\frac{pos}{10000^{2i/d_{\text{model}}}}\right)$$

- Alternativa: embedding posizionali appresi.

Perché il Transformer è adatto alla Traduzione

- **Parallelizzazione** $\rightarrow$ addestramento veloce ed efficiente.
- **Self-attention globale** $\rightarrow$ modella dipendenze a lungo raggio senza perdita di contesto.
- **Multi-head attention** $\rightarrow$ gestisce ambiguità sintattiche e scelte lessicali.
- **Architettura encoder–decoder** $\rightarrow$ naturale per input e output di lunghezza variabile.

Dataset – Dimensioni

- **Frasi totali:** 1.909.115
- **File paralleli:** `en.txt`, `it.txt`
- **Allineamento:** 1:1 (una frase EN una frase IT)

Table: Top 10 token per frequenza – EN vs IT

#	Token EN	Freq EN	Token IT	Freq IT
1	the	3,538,081	di	1,906,744
2	of	1,799,243	e	1,213,602
3	to	1,662,163	che	1,125,754
4	and	1,390,381	la	960,137
5	in	1,087,533	in	805,660
6	that	826,009	il	787,811
7	a	823,716	per	701,814
8	is	818,932	a	628,449
9	for	559,395	del	515,430
10	I	544,877	è	509,977

Lunghezze medie delle frasi

- **EN**:
  - Media: **26.05** token
  - Max: 668 token
- **IT**:
  - Media: **25.13** token
  - Max: 558 token
- $\Rightarrow$ Lunghezze bilanciate, ma presenza di outlier molto lunghi

Rapporto lunghezze IT / EN

- Calcolato per ogni coppia: $\frac{\text{len(IT)}}{\text{len(EN)}}$
- **Media:** 0.994
- **Mediana:** 0.966
- **Deviazione standard:** 0.616
- **Outlier:** max = 141.0 (!)

Strategia di Preprocessing – Overview

- Obiettivo: convertire file testuali paralleli IT/EN in batch tensorizzati, pronti per l'addestramento di un Transformer.
- Il preprocessing include:
  1. Caricamento delle frasi da file
  2. Costruzione dei vocabolari personalizzati
  3. Tokenizzazione numerica con 'bos', 'eos', 'unk'
  4. Padding per batching omogeneo
- Implementato interamente con PyTorch puro (no HuggingFace)

Caricamento frasi parallele

I due file ‘en.txt‘ e ‘it.txt‘ contengono frasi allineate riga per riga.

- Le liste ‘src_lines‘ e ‘tgt_lines‘ sono usate per:
  - costruire i vocabolari
  - creare il dataset
- Il vocabolario è costruito contando le frequenze
- Solo parole con frequenza $\geq min_{freq}$ vengono incluse.
- I token speciali pad, unk, bos, eos sono riservati ai primi 4 indici.

Tokenizzazione custom

- Ogni frase viene trasformata in una sequenza di indici:

```
class Tokenizer:
    def __init__(self, vocab): self.vocab = vocab
    def __call__(self, text):
        tokens = ['<bos>'] + text.split() + ['<eos>']
        return [self.vocab.get(t, self.vocab['<unk>']) \
                                    for t in tokens]
```

- Strategia molto semplice:
  - Split lessicale basato su ‘whitespace‘
  - Non case-insensitive
  - Non rimuove punteggiatura

Dataset e batching

- Dataset personalizzato PyTorch:

```
class TranslationDataset(Dataset):
    def __getitem__(self, idx):
        src_tokens = torch.tensor(src_tokenizer \
                                  (src_lines[idx]))
        tgt_tokens = torch.tensor(tgt_tokenizer \
                                  (tgt_lines[idx]))
        return src_tokens, tgt_tokens
```

- ‘collate_fn‘ gestisce il padding automatico:

```
def collate_fn(batch):
    src_batch, tgt_batch = zip(*batch)
    return pad_sequence(src_batch, padding_value=0), \
           pad_sequence(tgt_batch, padding_value=0)
```

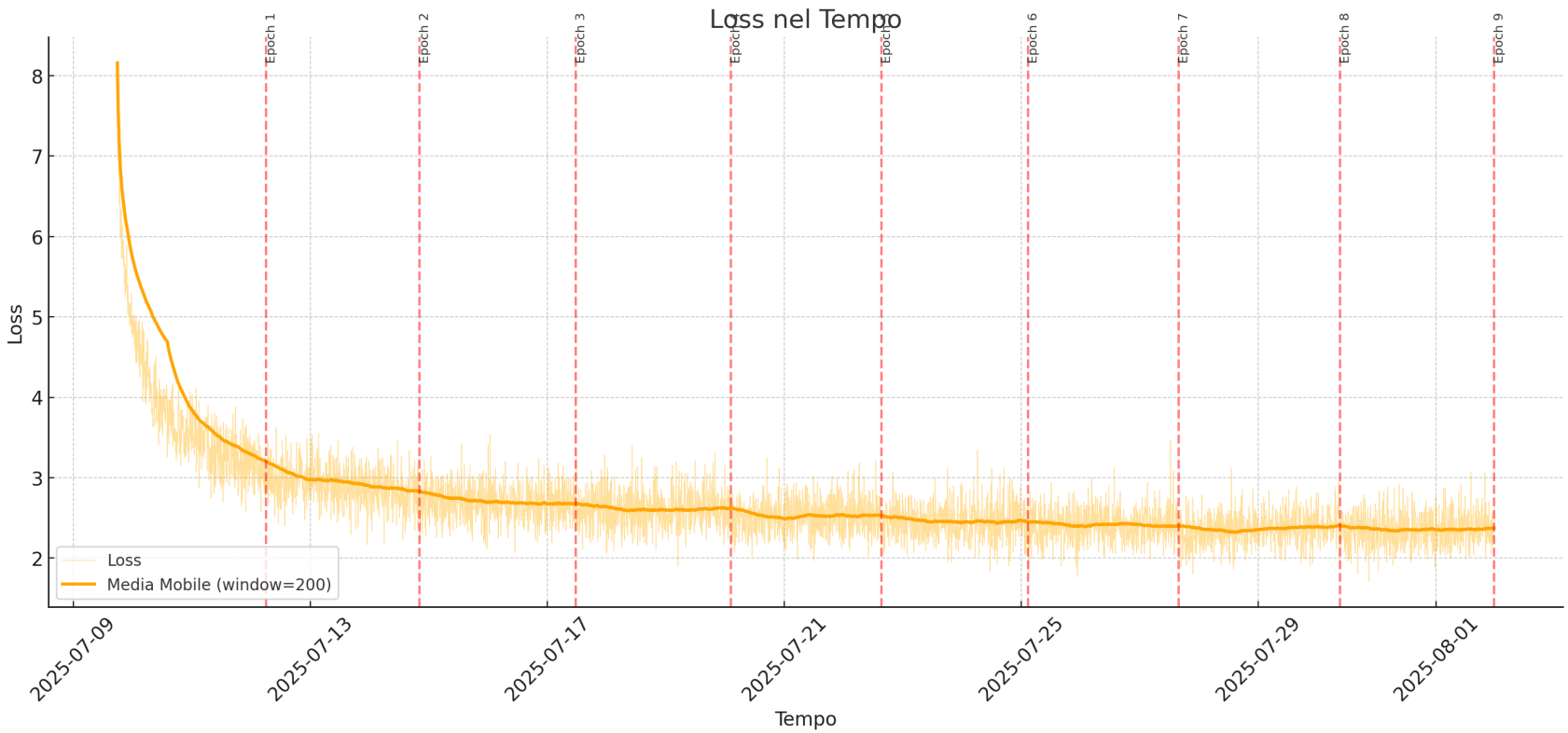
Scelte Infrastrutturali: Vincoli, Calcoli e Soluzioni 1/2

- **Fase iniziale: tentativo su Google Colab con GPU T4**
  Tokenizer allenato con successo, ma tempi limite imposti impedivano il training completo del modello sequenziale
- **Scelta alternativa: spostamento su server locale con sola CPU**
  Nessuna GPU disponibile, ma accesso illimitato in tempo $\rightarrow$ soluzione: training multithread su CPU
- **Calcolo stimato del tempo di training**
  - Batch size: 32 $\Rightarrow$ $\sim$60,000 batch per epoca
  - Velocità empirica: 100 batch / 6 minuti $\Rightarrow$ 1 batch 3.64 s
  - $\Rightarrow$ 1 epoca 60 ore, $\Rightarrow$ 10 epoche 600 ore ($\sim$25 giorni)

Scelte Infrastrutturali: Vincoli, Calcoli e Soluzioni 2/2

- **Problema: rischio di interruzioni e perdita di stato**
  Soluzione: salvataggio checkpoint a ogni epoca + log continuo delle performance
- **Checkpointing e logging**
  - Ogni checkpoint $\sim$3.5 GB, uno per epoca
  - Ogni checkpoint rappresenta un modello completamente addestrato a fine epoca i
  - File di log registrati in tempo reale (`tee` + `tail -f`)
- **Obiettivo finale**
  Confrontare le performance tra modelli salvati (1 per epoca) per selezionare il miglior compromesso tempo/qualità

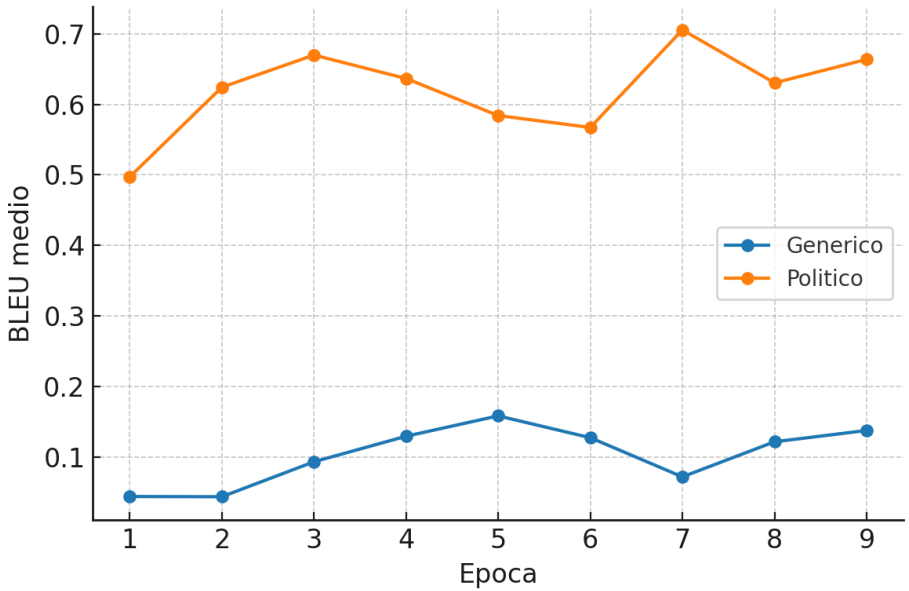
Loss nel Tempo



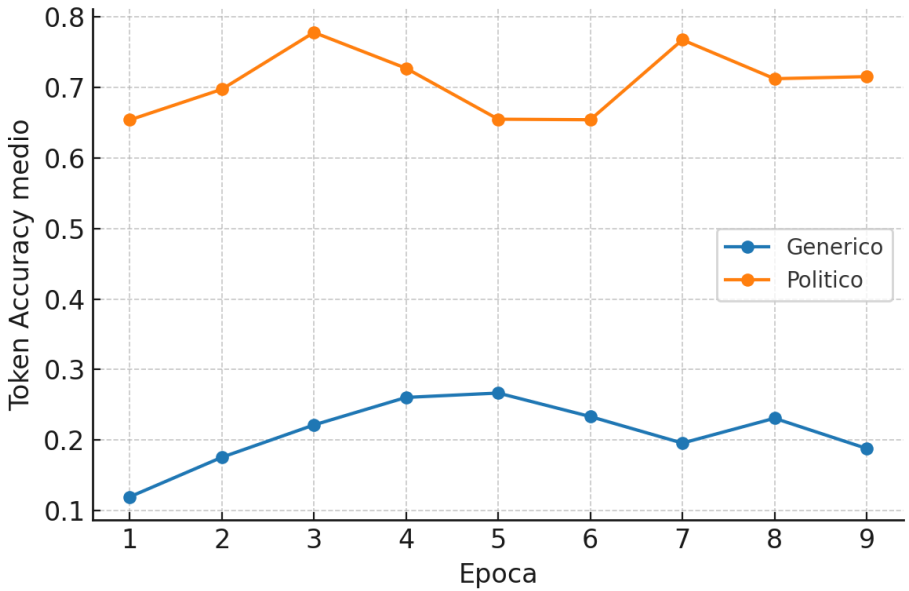
Metodo di testing

- **Set di frasi**: 20 frasi in totale
  - 10 *politiche* (in testa al file)
  - 10 *generiche* (in coda al file)
- **Modelli valutati**: 9 checkpoint (uno per epoca)
- **Metriche**:
  - **BLEU** corpus e per-dominio
  - **Token Accuracy** (percentuale token esatti)
- **Protocollo**:
  1. Inferenza su tutte le 20 frasi per ciascun checkpoint
  2. Calcolo metriche per epoca e per dominio (politico/generico)
  3. Confronto trend nel tempo per individuare il miglior checkpoint

Risultati complessivi nel tempo



BLEU medio per epoca (linea continua) + media mobile (linea smussata).



Token Accuracy per epoca (linea continua) + media mobile.

Nota: le curve sono calcolate su tutte le 20 frasi; dettagli per dominio nella slide seguente.

Frasi di test (non viste in training)

Politica (10)

1. The European Parliament will vote on the resolution next week.
2. We call for stronger democratic accountability in EU institutions.
3. The Commission has proposed a new directive on digital markets.
4. Foreign policy must remain consistent with the EU's core values.
5. The treaty was signed by all member states.
6. They debated climate neutrality strategies in the Strasbourg session.
7. The Council adopted new regulations on migration.
8. Freedom of expression is a pillar of European democracy.
9. Parliamentary sessions resumed after the summer break.
10. The new fiscal compact was met with mixed reactions.

Generico (10)

1. I forgot my keys at the restaurant.
2. She is learning to play the piano.
3. The weather today is sunny and warm.
4. My dog loves to run in the park.
5. Please bring a bottle of water for the hike.
6. I watched a great movie last night.
7. He is baking a cake for his mother's birthday.
8. We will meet at the train station at 6.
9. The book was better than the movie adaptation.
10. I enjoy painting landscapes during the weekend.

Politica — esempio con miglioramento marcato

The European Parliament will vote on the resolution next week.
Il Parlamento europeo voterà la risoluzione della prossima settimana.

Ep. 1	(BLEU 0.3303)	Il Parlamento europeo voterà sulla risoluzione della settimana prossima
Ep. 2	(0.3689)	Il Parlamento europeo voterà sulla risoluzione della prossima settimana.
Ep. 3	(0.6606)	Il Parlamento europeo voterà la risoluzione della prossima settimana.
Ep. 4	(0.3689)	Il Parlamento europeo voterà sulla risoluzione della prossima settimana.
Ep. 5	(0.5976)	Il Parlamento europeo voterà la prossima settimana.
Ep. 6	(0.5976)	Il Parlamento europeo voterà la prossima settimana.
Ep. 7	(0.5969)	Il Parlamento europeo voterà sulla risoluzione la prossima settimana.
Ep. 8	(0.5976)	Il Parlamento europeo voterà la prossima settimana.
Ep. 9	(0.4939)	Il Parlamento europeo voterà la prossima settimana sulla risoluzione.

Osservazione: netto miglioramento sintattico entro Ep.3; fluttuazioni successive sulla collocazione di *risoluzione*.

Politica — esempio già buono dal primo checkpoint

The Commission has proposed a new directive on digital markets.
La Commissione ha proposto una nuova direttiva sui mercati digitali.

Ep. 1 (BLEU 1.0000)	La Commissione ha proposto una nuova direttiva sui mercati digitali.
Ep. 2 (1.0000)	La Commissione ha proposto una nuova direttiva sui mercati digitali.
Ep. 3 (1.0000)	La Commissione ha proposto una nuova direttiva sui mercati digitali.
Ep. 4 (1.0000)	La Commissione ha proposto una nuova direttiva sui mercati digitali.
Ep. 5 (1.0000)	La Commissione ha proposto una nuova direttiva sui mercati digitali.
Ep. 6 (1.0000)	La Commissione ha proposto una nuova direttiva sui mercati digitali.
Ep. 7 (1.0000)	La Commissione ha proposto una nuova direttiva sui mercati digitali.
Ep. 8 (1.0000)	La Commissione ha proposto una nuova direttiva sui mercati digitali.
Ep. 9 (1.0000)	La Commissione ha proposto una nuova direttiva sui mercati digitali.

Osservazione: frase di dominio — perfetta e stabile lungo tutte le epoche.

Generico — esempio con miglioramento marcato

My dog loves to run in the park.
Il mio cane ama correre nel parco.

Ep. 1	(BLEU 0.0278)	La mia cane di fronte alla mia visita.
Ep. 2	(0.0843)	Il mio cane è stato costretto a gestire in occasione del 1980.
Ep. 3	(0.0251)	La mia cane è destinata a gestire nel panel .
Ep. 4	(0.0170)	I miei cane sono in grado di gestire il Golfo di seduta.
Ep. 5	(0.5874)	Il mio cane ama correre nel bacino del mare.
Ep. 6	(0.2778)	Il mio cane ama essere gestito nel bacino del Nord.
Ep. 7	(0.1275)	Il mio cane è chiamato a gestire nel Golfo.
Ep. 8	(0.0831)	Il cane ama il film nel sito.
Ep. 9	(0.4111)	Il mio cane ama andare in porto.

Osservazione: grande salto a Ep.5 su struttura SVO; persiste errore lessicale su *parco*.

Generico — controesempio che resta difficile

The weather today is sunny and warm.
Oggi il tempo è soleggiato e caldo.

Ep. 1 (BLEU 0.0393) La catastrofe è stata analizzata e chiara.
Ep. 2 (0.0603) Il clima di oggi è sconvolto e di sinistra.
Ep. 3 (0.0157) Oggi il maltempo è stato caratterizzato da una situazione di stallo, e di facile comprensione.
Ep. 4 (0.0251) Oggi il clima è stato messo in atto e preciso.
Ep. 5 (0.0467) Il clima è oggi viziato e piacevole.
Ep. 6 (0.0174) Il clima è stato caratterizzato da una situazione di degrado e di degrado dei costi.
Ep. 7 (0.0316) Il clima di clima è ormai difficile e stimolante.
Ep. 8 (0.0157) Oggi il clima è un bel po' di confusione e di angoscia per il momento.
Ep. 9 (0.0188) Il clima è un clima di calma, di incertezza e di prezzi più brevi.

Osservazione: frase semplice ma fuori dominio; deriva semantica persistente $\rightarrow$ scarsa generalizzazione su frasi generiche.

Conclusioni

- **Punti di forza:**
  - Capacità di apprendere e tradurre correttamente termini e strutture mai viste, anche da contesti minimi (*zero-shot* su lessico generico).
  - Risultati stabili e accurati su alcune frasi di dominio politico già dalle prime epoche.
  - Miglioramenti rapidi (entro Ep.3–5) in frasi complesse quando il lessico è parzialmente noto.
- **Criticità:**
  - Prestazioni altalenanti su frasi politiche con strutture sintattiche ambigue o collocazioni lessicali variabili.
  - Scarsa capacità di generalizzare in contesti fuori dominio (*out-of-domain*), anche su frasi semplici.
  - Errori semantici persistenti dovuti a inferenza contestuale insufficiente.
- **Prospettive di miglioramento:**
  - Ampliare il dataset con esempi bilanciati tra dominio politico e generico.
  - Adottare tecniche di *domain adaptation* e *data augmentation* mirata.
  - Sperimentare con *fine-tuning* su frasi fuori dominio per aumentare la robustezza semantica.

Sviluppi futuri

Dati & tokenizzazione

- Passaggio a subword (*BPE*/Unigram, byte-level), vocabolario condiviso EN–IT, truecasing & normalizzazione.
- Bilanciamento domini: campionamento controllato (politico vs generico), *curriculum learning*.

Training

- Scheduler con warmup + *inverse-sqrt*, *label smoothing*, dropout, gradient clipping.
- *Early stopping* su dev set + *checkpoint averaging*; se disponibile GPU: mixed precision.
- *Adapters*/LoRA per *domain adaptation* senza costi full fine-tune.

Decoding

- Tuning di beam size, *length normalization* e coverage penalty.
- *Constrained decoding* con glossari (terminologia istituzionale), *lexicon biasing*; *shallow fusion* con LM.

Data-centric

- Back-translation da monolingue (IT & EN) e *noising* controllato per robustezza out-of-domain.
- Set di validazione dedicati per: frasi politiche, frasi generiche semplici, frasi lunghe/outlier.

Piano operativo & metriche / valore

Valutazione

- Oltre BLEU/accuracy: chrF, COMET; report per-dominio (politico vs generico) e per-famiglia sintattica.
- Challenge set mirati (lessico istituzionale, collocazioni ambigue); test di significatività tra checkpoint.
- Human eval (MQM) su campione: adeguatezza, fluency, terminologia.

Deployment & MLOps

- API di inferenza + batch; detokenizzazione stabile; caching risultati ripetuti.
- Monitoraggio in produzione: drift lessicale, tassi di post-edit, *QE* per prioritizzare la revisione umana.

Linee di business (upsell)

- Motore NMT *domain-adapted* per PA/EU; gestione glossari e stile istituzionale.
- Servizi di *fine-tuning* su dati del cliente; pipeline di *back-translation* proprietaria.
- Quality Estimation + reportistica SLA (riduzione costi di post-edit).