Summarization con Transformers

Nel corso della mia attività progettuale ho scelto di dedicarmi all'ambito del Natural Language Processing (NLP), in particolare ha destato in me molto interesse dal primo momento l'ambito della Summarization.

Enhancing a Text Summarization System with Elmo

Mi è stato proposto di studiare il paper della fine del 2019 *Enhancing a Text Summarization System with ELMo, Mastronardi e Tamburini* e a partire da questo cercare di migliorare i risultati ottenuti su alcuni task di summarization sfruttando delle tecnologie più evolute come i Transformers, in particolare BERT.

Nel Paper in questione viene sfruttata una rete Pointer-Generator e gli esperimenti con essa sono condotti su due datasets: Il primo è il dataset CNN/DailyMail, il quale contiene oltre 300 mila entries formate da articoli scritti da giornalisti e scaricati tramite web scraping dal sito cnn.com. Il secondo dataset è il dataset Newsroom, formato da 1.3 milioni di coppie articoli-sintesi e creato appositamente per il task di summarization.

La rete sopra citata consiste in un'architettura encoder-decoder. L'encoder e il decoder sono basati su una rete RNN Long-Short-Term-Memory (LSTM) a singolo strato. Tuttavia, la rete Pointer-Generator originale non sfrutta il Transfer Learning, al contrario in questo lavoro vengono utilizzati dei word emeddings pre-trained, nello specifico si fa ricorso ad ELMo (Embedding from Language Model) il quale consiste in degli embeddings formati in funzione dell'intera sequenza di ingresso e di conseguenza ritenuti particolarmente funzionali per il task in oggetto.

Le metriche

I risultati ottenuti sono stati misurati con ROUGE (Recall-Oriented Understudy for Gisting Evaluation), questo è un insieme di metriche e pacchetti software finalizzati alla valutazione dell'automatic summarization e della machine translation. Le metriche confrontano le sintesi o le traduzioni prodotte in modo automatico da un modello con delle sintesi o traduzioni di riferimento prodotte manualmente da persone.

Nello specifico sono state prese in considerazione 3 metriche:

- ROUGE-1 (R1), la quale misura la sovrapposizione tra gli unigrammi (le singole parole) prodotti dal modello con quelli delle sintesi di riferimento;
- ROUGE-2 (R2), la quale misura la sovrapposizione tra i bigrammi (le coppie di parole) prodotti dal modello con quelli delle sintesi di riferimento;

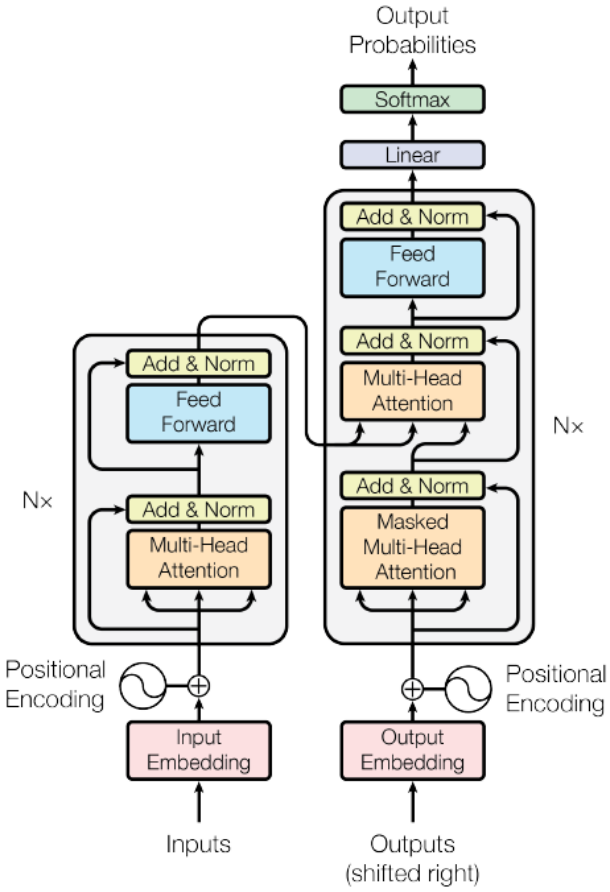
- ROUGE-L (RL), la quale misura delle statistiche basate sulle più lunghe sotto sequenze in comune tra le sintesi prodotte dal modello e quelle di riferimento.

I Risultati ottenuti dal modello Pointer-Generator basato sugli embeddings di ELMo nelle metriche appena citate sul task CNN/DailyMail sono:

- R1 = 38.96
- R2 = 16.25
- RL = 34.32

I Transformers

L'Attention, nell'ambito del machine learning, è una tecnica volta a riprodurre il processo cognitivo di attenzione degli esseri umani. Lo scopo è quello di focalizzarsi sulle parti importanti contenute in un input a discapito del resto, ciò implica la necessità di concentrare grandi risorse computazionali su piccole parti dei dati in ingresso. L'importanza di un dato è dettata dal contesto.



Il meccanismo di Attention è da tempo utilizzato con successo nell'ambito del Natural Language Processing (NLP) e nella computer vision, ma è dal 2017, con l'introduzione dell'architettura dei

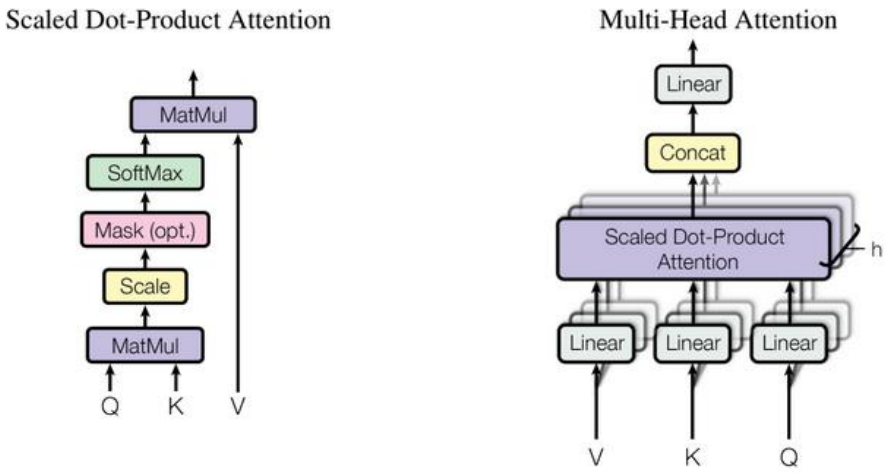
Transformers, presentata nel Paper *Attention is all you need, Vaswani et al.*, che l'attention ha visto incrementare il suo utilizzo in modo massiccio, diventando una parte integrante di diverse architetture impiegate nei tasks più disparati.

Nei Transformers, come in altri modelli di reti neurali performanti, viene utilizzata una struttura di Encoder e Decoder. L'Encoder si occupa di codificare una sequenza di simboli in ingresso in una sequenza con rappresentazione continua, questa sequenza sarà poi nuovamente tradotta dal Decoder che la riporterà in una condizione simile a quella di partenza.

L'Encoder, rappresentato nella parte sinistra dell'immagine superiore, è formato da una pila di strati identici. Ogni strato ha due sotto strati, il primo è un meccanismo di self-attention multi-head, il secondo è invece un semplice strato di propagazione posizionale.

Il Decoder, rappresentato nella parte destra dell'immagine superiore, ha una struttura simile a quella dell'Encoder, l'unica differenza è l'aggiunta di un terzo sotto strato, il quale applica il meccanismo di multi-head attention all'output della pila di Encoder.

Una funzione di Attention si occupa di tradurre l'insieme di una query e un set di coppie chiave-valore in un output, a patto che tutti i dati in gioco siano sotto forma di vettori. L'output viene calcolato come la somma pesata dei valori e i pesi di ogni valore sono calcolati sulla base di una funzione di compatibilità della query con la corrispondente chiave.



L'Attention riportata nella figura precedente e utilizzata nell'architettura originale dei Transformers è definita "Scaled Dot-Product Attention" in quanto viene calcolato il prodotto scalare delle query (Q) con tutte le chiavi (K), entrambe di dimensione d_k e successivamente viene "scalato" dividendo il tutto per un fattore uguale a $\sqrt{d_k}$, solo a questo punto si applica una funzione softmax per calcolare i pesi dei valori in ingresso con dimensione d_v.

La peculiarità dell'architettura dei Transformer risiede nell'applicazione in parallelo della funzione di attention (Multi-head Attention); questo è possibile proiettando linearmente le queries, le chiavi e i valori h volte e calcolando sulle proiezioni l'attention, ottenendo così un valore in output di dimensione d_v , questi vengono quindi concatenati e linearizzati nuovamente per ottenere il valore finale.

BERT

Bert è un Language representation Model, il suo nome è l'acronimo di Bidirectional Encoder Representations from Transformers. La potenza di questo modello e la grande diffusione è giustificata dalla sua riutilizzabilità, partendo dall'architettura originale è possibile effettuare agevolmente un'operazione di fine-tuning sui task più disparati a seconda delle proprie esigenze ed ottenere risultati vicini o superiori allo stato dell'arte dei modelli suoi predecessori.

La sua struttura è pressochè identica a quella dei Transformers illustrata in precedenza.

Summarization

La Summarization, ovvero il processo di sintesi di un testo, può essere categorizzata secondo strategie extractive oppure abstractive. Per Extractive si intende il meccanismo di selezione delle migliori N frasi capaci di rappresentare al meglio il concetto espresso in un testo. Per Abstractive invece si intende la capacità del modello di rappresentare i punti chiave del discorso utilizzando parole e frasi differenti. È evidente come il secondo risulti essere più complesso, esso implica infatti che il modello sia in grado di "capire" un concetto al punto da estrarne i passaggi chiave e successivamente avere carattere generative, ovvero la capacità di generare di sua sponte un output riformulato del testo in ingresso.

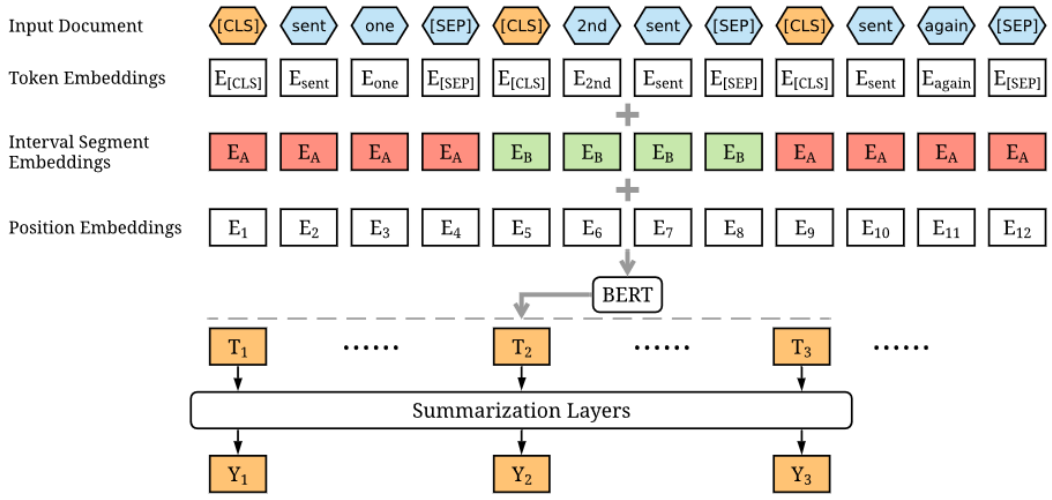
La metrica più diffusa per misurare l'efficacia di un modello in ambito summarization è Rouge, la quale misura la capacità del modello di riprodurre sintesi il più simili possibile, in termini di sovrapposizione degli n-grammi, a delle sintesi fornite come standard in fase di training.

Utilizzare BERT per Summarization

Ad un primo sguardo, nonostante la potenza di BERT, sembrerebbe un modello non adatto alla summarization, la sua struttura bidirezionale, infatti, è un ostacolo visto che per task di tipo generative

sarebbe necessario che il modello fosse in grado di campionare le parole precedentemente generate e tramite una distribuzione di probabilità capire quale parola generare successivamente.

Tuttavia, in letteratura sono presenti diversi esempi di utilizzo di BERT in ambito summarization. BERTSum è un modello descritto nel paper del Settembre 2019 Fine-tune *BERT for Extractive Summarization, Liu;* Questo modello, di cui è disponibile il codice open-source su Git-hub, è in grado di effettuare una summarization di tipo extractive. BERT è creato come modello masked-language, di conseguenza ragiona per token e non per frasi; per questo motivo è stato necessario modificare le sequenze in input, aggiungendo dei token separatori manualmente tra le diverse frasi come è possibile vedere nell'immagine seguente.



Dal nostro punto di vista, è però di maggiore interesse la Summarization di tipo abstractive. Una soluzione a tale problema, in grado di utilizzare BERT, è il modello EncoderDecoder implementato da HuggingFace e descritto nel paper *Leveraging Pre-trained Checkpoints for Sequence Generation Tasks di Rothe et al., 2020*.

Questo modello presenta un modello di tipo sequence-to-sequence basato sui Transformers. Questo significa che è possibile sfruttare i checkpoint di pressochè qualsiasi Transformer pre-trained disponibile per completare task di carattere generative, ma non solo.

Nel nostro caso, risultano di particolare interesse i risultati ottenuti da questo genere di modello nell'ambito della Summarization abstractive riportati nella tabella seguente.

	Gigaword			CNN/Dailymail			BBC XSum		
	R-1	R-2	R-L	R-1	R-2	R-L	R-1	R-2	R-L
Lead	–	–	–	39.60	17.70	36.20	16.30	1.61	11.95
PtGen	–	–	–	39.53	17.28	36.38	29.70	9.21	23.24
ConvS2S	35.88	17.48	33.29	–	–	–	31.89	11.54	25.75
MMN	–	–	–	–	–	–	32.00	12.10	26.00
Bottom-Up	–	–	–	41.22	18.68	38.34	–	–	–
MASS	*38.73*	19.71	*35.96*	–	–	–	–	–	–
TransLM	–	–	–	39.65	17.74	36.85	–	–	–
UniLM	–	–	–	*43.47*	*20.30*	*40.63*	–	–	–
Initialized with the base checkpoint (12 layers)									
RND2RND	36.94	18.71	34.45	35.77	14.00	32.96	30.90	10.23	24.24
BERT2RND	37.71	19.26	35.26	38.74	17.76	35.95	38.42	15.83	30.80
RND2BERT	37.01	18.91	34.51	36.65	15.55	33.97	32.44	11.52	25.65
BERT2BERT	38.01	19.68	35.58	39.02	17.84	36.29	37.53	15.24	30.05
BERTSHARE	38.13	***19.81***	**35.62**	39.09	18.10	36.33	38.52	16.12	31.13
ROBERTASHARE	**38.21**	19.70	35.44	**40.10**	**18.95**	**37.39**	**39.87**	**17.50**	**32.37**
GPT	36.04	18.44	33.67	37.26	15.83	34.47	22.21	4.89	16.69
RND2GPT	36.21	18.39	33.83	32.08	8.81	29.03	28.48	8.77	22.30
BERT2GPT	36.77	18.23	34.24	25.20	4.96	22.99	27.79	8.37	21.91
ROBERTA2GPT	37.94	19.21	35.42	36.35	14.72	33.79	19.91	5.20	15.88
Initialized with the large checkpoint (24 layers)									
BERTSHARE	38.35	19.80	35.66	39.83	17.69	37.01	38.93	16.35	31.52
ROBERTASHARE	**38.62**	19.78	**35.94**	**40.31**	18.91	**37.62**	***41.45***	***18.79***	***33.90***

La tabella in questione riporta i risultati ottenuti, su diversi dataset, nelle metriche ROUGE spiegate in precedenza, da svariati modelli sviluppati negli anni e capaci di effettuare Summarization; tra questi, nella colonna centrale è visibile il task CNN/Dailymail materia di studio di questa attività progettuale.

Nella parte superiore della tabella sono riportati modelli non basati su Transformers, nella parte centrale (sotto la dicitura “Initialized with the base checkpoint”) sono riportati i modelli EncoderDecoder inizializzati con i checkpoint di alcuni Transformers (GPT-2, BERT, RoBERTa) nelle loro versioni più leggere formate da meno parametri, ad esempio 110M di parametri per bert-base rispetto ai 330M di bert-large o inizializzati in modo random come nel caso del modello RND2RND. Nella parte inferiore infine è possibile vedere i risultati ottenuti dai modelli large.

Per i nostri scopi, tra tutti quelli mostrati, è stato preso in esame il modello basato esclusivamente su BERT sia per l’encoder che per il decoder, ed in particolare il modello base per una questione di risorse computazionali a disposizione.

Nello specifico è quindi possibile vedere come il modello BERT2BERT (evidenziato in tabella) con 12 layer ed inizializzato con i checkpoint della versione base ha ottenuto risultati migliori rispetto al modello basato su ELMo contenuto nel paper preso in esame per questo progetto. L'obiettivo a questo punto è stato quindi cercare di replicare i risultati ottenuti dal modello BERT2BERT implementando autonomamente questo modello.

Implementazione

Per implementare il modello è stata utilizzata la piattaforma Colab, gratuitamente fornita da Google e in grado di mettere a disposizione di ogni utente una GPU da 12GB.

Il codice scritto è basato sulla suite HuggingFace, la quale mette a disposizione una mole infinita di codice già pronto:

- checkpoint di modelli Transformer molto performanti e comodi da utilizzare;
- una varietà di DataSet facilmente scaricabili, compreso quello su cnn/dailymail di nostro interesse;
- centinaia di metriche per misurare le performance del nostro modello, compresa ROUGE e METEOR utilizzate in questo lavoro.

METEOR (Metric for Evaluation of Translation with Explicit ORdering) è una metrica nata per valutare la qualità degli output in ambito machine translation, tuttavia, risulta essere applicabile anche alla summarization. È basata sulla media armonica di Precision e Recall degli unigrammi. A differenza delle metriche ROUGE spiegate in precedenza, METEOR non si limita a misurare il combaciare delle parole esatte, ma applica anche meccanismi di stemming (riduzione di parole derivate alla propria radice) e sinonimia.

La versione originale del dataset CNN/Dailymail non era pensata per il task di Summarization, era invece utilizzata nell'ambito della machine reading e comprehension oltre che nel question answering. Successivamente sono state rilasciate le versioni 2.0.0 e 3.0.0, le quali hanno reso possibile il suo utilizzo in ambito Summarization cambiandone la struttura. In queste versioni gli articoli sono collegati ad una o più frasi scritte dagli autori degli articoli stessi come punti salienti del discorso. L'unica differenza tra le versioni 2 e 3 di questo dataset consiste nel fatto che nella più recente non è più presente l'anonimizzazione di cose o persone. Le metriche ROUGE e METEOR sono quindi calcolate confrontando le sintesi prodotte dal modello con le frasi salienti scritte dagli autori degli articoli originali.

Sono stati prodotti due notebook differenti, il primo (consultabile al seguente link: https://colab.research.google.com/drive/19HAyiWhICp-M6rqpf5msywetapm9Z-_b?usp=sharing), contiene il codice necessario per effettuare il training di un nuovo modello avente come Encoder e come Decoder dei modelli bert-base-uncased, tuttavia, il training di questo genere di modello richiede diverse ore di training e risorse di GPU ingenti per ottenere dei risultati soddisfacenti, per questo motivo è stato riportato solo a scopo esemplificativo un mini-training del modello, effettuato sfruttando una minima parte del dataset CNN/DailyMail, e facilmente modificabile sostituendo alcuni parametri a seconda delle proprie esigenze, come descritto all'inizio del notebook stesso.

Il secondo notebook invece, (consultabile qui: https://colab.research.google.com/drive/10AtuoraHPuOwDl-TUuKgUzJbhEg0chhB?usp=sharing) contiene il codice per verificare i risultati ottenuti da un modello Bert2Bert pre-trained sul dataset cnn/dailymail e poterlo quindi paragonare ai risultati ottenuti con ELMo. Il training di questo modello è stato svolto sul training set della versione 3.0.0 del dataset CNN/DailyMail, il quale contiene 286817 articoli con una lunghezza media di 766 parole distribuite in 29.74 frasi ed altrettante sintesi corrispondenti formate in media da 53 parole distribuite in 3.72 frasi. I risultati ottenuti sulle metriche ROUGE e METEOR fanno riferimento al test set di CNN/DailyMail 3.0.0 formato da 11487 coppie di articoli e sintesi non sottoposte al modello in fase di training.

Conclusioni

Risulta evidente come BERT, e più in generale i Transformers, siano una tecnologia dirompente e in grado di settare lo stato dell'arte in quasi ogni task del NLP ed image processing. Il caso studio preso in esame non è da meno, il modello BERT2BERT addestrato per effettuare abstractive summarization sul dataset cnn/dailymail è in grado di ottenere risultati migliori di 0.97 punti su R1 e 1.97 punti su R2 se confrontati con il modello basato su ELMo.

Un possibile limite dei Transformers come BERT risiede nelle ingenti risorse richieste per processare i dati, al crescere della lunghezza degli input cresce quadraticamente il consumo di memoria. Una possibile soluzione e sviluppo futuro potrebbe essere l'architettura di Performer, un sistema diverso di concepire il meccanismo di Attention in grado di rendere lineare il consumo di risorse rispetto alla lunghezza degli input e teoricamente compatibile con ogni tipo di Transformer, esiste un'implementazione di questo tipo di Attention anche in HugginFace, incapsulata all'interno del modello BERT stesso.