Università di Pisa, Human Language Technologies

Question Generation con BERT

Andrea Rosasco

May 29, 2020

1 Introduzione

La Neural Question Generation (NQG) è il campo, complementare al Question Answering, che si occupa della generazione di domande pertinenti ad un dato input. La generazione di domande trova applicazioni pratiche in diversi contesti: valutazione di studenti, tutoraggio adattivo o generazione di dataset.

Il tipo di question generation implementata per il progetto, consiste nella generazione di domande, dato il testo e la risposta. La domanda generata dovrà avere come risposta, quella desiderata.

Nel progetto, ho provato ad utilizzare le capacità di language understanding del modello BERT [8], come punto di partenza nella creazione di un sistema di question generation.

Il modello finale riesce a creare domande ben formulate relative al testo in input. Non è tuttavia chiaro quante di queste abbiano come risposta quella passata in input.

2 Materiali e Metodi

Il modello utilizzato consiste in un'architettura encoder-decoder, dove encoder e decoder sono rispettivamente BERT ed una Gated Recurrent Unit (GRU), con attention implementata come descritto in [3].

BERT, come spiegato in modo più approfondito in seguito, è stato utilizzato in due modi diversi. Prima ho provato l'approccio "out-of-the-box", non includendolo quindi nel ciclo di training. In seguito, ho provato la two-stage optimization descritta in [2] che prevede uno stage iniziale in cui solo il decoder è allenato ed un secondo stage nel quale l'encoder viene incluso nel training loop.

Per incrementare le performance del modello, misurate tramite BLEU score, si è provato a generare l'output tramite Beam Search, ma l'approccio non ha portato a nessun sensibile miglioramento.

Il training del modello è stato eseguito con i seguenti parametri:

- Optimizer: Adam
- Learning Rate: 1e-4
- Dropout: 0.5
- Teacher Forcing: 0.5
- Loss: Cross Entropy
- Evaluation: BLEU

Ho provato ad utilizzare due diverse versioni di Bert: la versione "base" (12 layer, hidden size 768 , 12 attention head, 110ML di parametri) e la versione "large" (24 layer, hidden size 1024 , 16 attention head, 340ML di parametri). La versione "large" si è rivelata però molto lenta per

il training. In parte per il numero elevato di parametri da aggiornare ad ogni step ed in parte perché, per questioni di memoria, è stato necessario usare mini-batch di dimensione ridotta, rendendo soprattutto problematica la two-stage optimization (oltre ai parametri viene salvato in GPU anche il gradiente). Per questi motivi, e per il fatto che l'utilizzo della versione "large" non fornisse un sostanziale aumento delle prestazioni, ho scelto di svolgere gli esperimenti successivi utilizzando la versione "base".

Sempre per ragioni di performance ho scelto i seguenti parametri per il decoder

- input embedding = 512
- hidden size = 512
- attention size = 512

Per un totale di 70ML di parametri.

Il training è stato svolto su GPU, utilizzando una delle schede video Tesla V100 del dipartimento. Come libreria è stata utilizzata PyTorch insieme alla libreria "transformers" di Hugging Face che mette a disposizione un implementazioni di BERT.

3 Data Preprocessing

Il primo passo è stato estrarre da SQUAD tutte le possibili triple (context, answer, question) dove l'input del modello è dato da x = (context, answer) e l'output/ground truth è y = question. Dopo aver adeguatamente preparato context e answer (con padding e tokenizzazione), li ho concatenati, mantenendoli però separati tramite uno specifico token (i.e. [SEP]) e salvati in un file insieme alle ground truth.

4 BERT

Solitamente, a causa delle enormi risorse necessarie per il training di BERT, il modello si inizializza con i pesi forniti da Google. Esistono diversi insiemi di pesi che variano per dimensione del modello e del vocabolario utilizzato. Nel progetto ho usato la versione 'base-cased', così da sfruttare appieno la presenza della maiuscola come caratteristica aggiuntiva nel processo di language understanding.

Una volta scaricati modello e pesi, possiamo utilizzare BERT in due modi diversi: quello feature based e quello basato su fine-tuning.

Facendo fine-tuning, BERT viene attivamente allenato durante il training. Nonostante questo approccio possa portare a performance migliori è anche molto più costoso a livello computazionale, in quanto ad ogni epoch dovremmo passare l'input a BERT, calcolare il risultato e aggiornarne i pesi.

L'approccio feature based è statico e consiste semplicemente nell'escludere BERT dal ciclo di traingin ed usarlo solamente per calcolare gli encoding dell'input. Ovviamente in questo modo BERT non si specializza nel task per cui sta venendo utilizzato, ma le capacità di language understanding "out-of-the-box" date dal pretraining sono comunque ottime.

Ho deciso di testare entrambi gli approcci ma, essendo il fine-tuning di per sè troppo costoso, l'ho implementato nel contesto della two-stage optimization.

Gli encoding contestuali dell'input possono essere "estratti" da BERT in diversi modi: si possono concatenare tutti gli hidden state, farne la media, farne la somma pesata, etc. L'approccio scelto, testato anche nella pubblicazione originale, è quello di utilizzare solamente l'ultimo hidden state per ogni parola.

5 GRU con Attention

La scelta del modello per la generazione dell'output è ricaduta su di una GRU (Gated Recurrent Unit) in quanto pre-

senta performance simili ad una LSTM (Long Short-Term Memory) ma ha costo computazionale e dimensone inferiori.

Normalmente in un'architettura seq2seq, dove entrambi i componenti sono RNN, il primo hidden state in input al decoder è l'ultimo generato dall'encoder, in quanto contiene informazione sull'intera frase. In questo caso l'encoder non è una RNN e gli hidden state che genera sono relativi solo alla corrispondente parola in input. Anche in questo caso esistono diverse strategie di azione. Si può calcolare il primo hidden state passando l'output di BERT attraverso un fully-connected layer o si può inizializzare il primo hidden state a zero. Visto le dimensioni già consistenti del modello ho optato per la seconda alternativa.

Per condizionare l'output del decoder sugli encoding delle diverse parole in input ho utilizzato il meccanismo di attention presentato per la prima volta in (Bahdanau et al. 2015)[3]. Tale meccanismo consiste nel concatenare la query (in questo caso l'hidden state della GRU allo step precedente) ad ognuno dei valori (gli encoding generati da BERT) e passare i vettori così ottenuti in una feedforward neural network allenata insieme al resto del modello. L'output layer della rete ha come funzione di attivazione una softmax, i cui valori sono usati per calcolare la somma pesata degli encoding, ottenendo il cosiddetto attention vector. Si noti che l'attention vector non è altro che una somma pesata degli encoding generati da BERT ed ha la loro stessa dimensione.

Ad uno step qualsiasi del processo di training la GRU prende in input con probabilità 1/2 la parola generata allo step precedente o la parola esatta dalla ground truth (teacher-forcing). La parola è sostituita dal suo encoding (inizializzato randomicamente e allenato insieme al modello) che è passato, insieme all'attention vector, in input alla GRU, la quale, a sua volta, restituisce in output l'hidden state attuale.

A questo punto hidden state attuale, attention vector e input vengono concatenati. Il vettore così ottenuto viene infine passato attraverso un softmax layer il cui output è un vettore della dimensione del vocabolario (i.e. 28,996 parole) contenente per ogni parola la corrispondente probabilità associata.

6 Training

La loss di una sequenza viene calcolata come la somma delle cross entropy loss tra le parole generate e quelle corrette. Dopo aver calcolato le loss di ogni sequenza in una data minibatch ne viene fatta la media. A questo punto calcoliamo il gradiente e aggiorniamo i pesi tramite l'algoritmo di ottimizzazione Adam.

Durante il training sono stati utilizzati teacher-forcing con valore 0.5, il cui funzionamento è stato descritto nella sezione precedente, e dropout, sempre con valore 0.5.

6.1 Featue-based

Per il training feture-based sono riuscito ad utilizzare minibatch di dimensione 88. Questo ha portato a tempi di esecuzione relativamente brevi e mi ha permesso di continuare il training fino alla convergenza del modello, raggiunta dopo 30 epoch.

6.2 Two-stage

Per il two-stage training sono partito da checkpoint generati dal training feature based, precedentemente salvati, ed ho continuato il training per una decina di epoch propagando il gradiente fino a BERT.

In questo caso ho dovuto scegliere minibatch di dimensione molto inferiore (i.e. 14) portando questa fase ad essere molto più lenta.

7 Testing

Per il testing abbiamo misurato il BLEU score, provando a generare le frasi sia tramite un approccio greedy che tramite beam search (ovviamente disattivando dropout e teacher-forcing).

Ho scelto il BLEU score in quanto la Cross Entropy Loss non sembrava misurare efficacemente il risultato sul validation set, aumentano invece di diminuire. Il motivo è che per ottenere una buona Cross Entopy Loss il modello dovrebbe generare esattamente la stessa frase ma, nella realtà, esistono più modi di formulare la stessa domanda. Il BLEU score presenta una valida alternativa ma, per una misurazione più precisa doveremmo avere a disposizione formulazioni alternative della stessa domanda (i.e. dataset ad-hoc per question generation)

Le misurazioni sono state fatte su un validation set contenente il 25% dei pattern di SQuAD. Durante lo splitting del dataset sono stato attento ad evitare che le tuple (context, question, answer) del validation set non contenessero valori di "context" già presenti o collegati a quelli di tuple contenute nel training set: essendoci più domande con lo stesso context ma anche context collegati fra loro (i.e. estratti dalla stessa pagina di wikipedia) una divisione casuale avrebbe potuto gonfiare il validation score.

8 Risultati

8.1 Featue-based

Come si nota dal grafico in Figura 1, il modello converge nelle le prime 30 epoch, dopo le quali la loss smette di diminuire in modo significativo. Similmente il BLEU score raggiunge il suo picco intorno alle 30 epoch e inizia a diminuire tra le 30 e le 40 epoch.

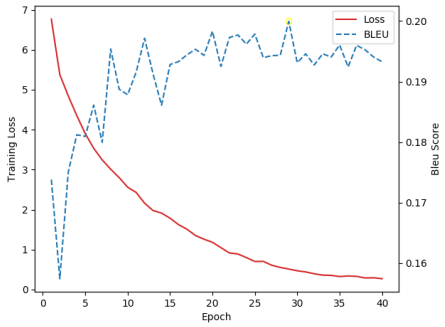

Figura 1: Traingin Loss e Validation BLEU Score del modello allenato secondo l'approccio feature-based

Più precisamente il massimo bleu score è ottenuto all'epoch 29 ed ha valore 19.95.

8.2 Two-stage

Come si può vedere dal grafico in Figura 2, la two-stage optimization non è riuscita a migliorare i risultati, al contrario, li ha peggiorati. Quando si include BERT nel ciclo di training abbiamo un rapido aumento della loss ed una corrispondente diminuzione del BLEU score. Continuando il training il modello converge ad un minimo locale superiore a quello precedentemente trovato portando a risultati peggiori. Una possibile spiegazione sta nel fatto che per includere BERT nel training loop abbiamo dovuto diminuire la dimensione delle minibatch a 14. Il gradiente calcolato su 14 pattern invece che su 88 è probabilmente più rumoroso e punta ad un diverso minimo locale. In 10 epoch quindi il modello esce dal minimo calcolato in precedenza e si sposta verso quello con loss più alta.

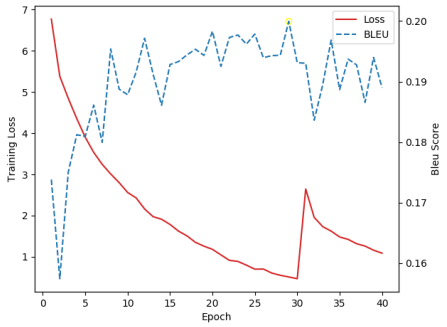

Figura 2: Traingin Loss e Validation BLEU Score del modello allenato tramite two-stage optimization

8.3 Beam Search

Un altro risultato peculiare, consiste nel fatto che, il tentativo di miglioramento dei risultati mediante l'applicazione dell'algoritmo di beam search, non fornisca un miglioramento del BLEU score. Con ampiezza 2 infatti otteniamo un score di 19.84, mentre con ampiezza 1, equivalente alla greedy search, lo score è di 19.95. Una possibile interpretazione è che, nonostante vengano considerate due possibili domande, il più delle volte quella più probabile sia la stessa che verrebbe generata da un approccio greedy; nei restanti casi in cui è la seconda alternativa a vincere, questa ha BLEU score minore.

Conclusioni

La preparazione del progetto è stata interessante sotto diversi punti di vista. Prima di tutto, ha evidenziato l'importanza della fase di analisi e la necessità di creare solide basi teoriche, con la lettura di pubblicazioni correlate, prima dello sviluppo. Inoltre, ha costituito il mio primo approccio al training con GPU e all'utilizzo di modelli con milioni di parametri. In questo scenario ho appreso come, a causa dei tempi di training esageratamente lunghi, l'approccio trial and error non sia un opzione e di come sia necessaria particolare attenzione nella fase di set-up.

Nonostante i punteggi ottenuti non siano comparabili con lo state-of-the-art, i risultati sono buoni se si considera la semplicità del modello. Tali risultati sono ascrivibili alla capacità di comprensione del testo di BERT.

9 Ulteriori Sviluppi

Vorrei elencare in questa sezione possibili miglioramenti al progetto, che non ho potuto testare. Consiglio, nel caso ne si volesse provare l'implementazione, di essere provvisti di una macchina con sufficiente potenza di calcolo. Nonostante Google Colab fornisca gratuitamente una GPU in un ambiente di sviluppo interattivo, non è in alcun modo da considerarsi ottimale, in quanto non pensato per uso intensivo. Presenta limiti sul tempo di computazione ed è più volte risultato instabile nel trasferimento di grandi dataset. I possibili miglioramenti sono i seguenti:

- Implementazione di un meccanismo per la generazione di parole rare (non presenti nel vocabolario), e.g. Pointer Softmax [4] Hybrid Architecture (word level + character level), etc.
- Valutazione dei risultati tramite modello di Question Answering [6]
- Ottimizzazione del BLEU score
- Utilizzo di una Beam Search differenziabile durante il training
- Sostituzione della RNN con il decoder di un Transformer [5] (fine tuning di GPT2 condizionato su BERT?) [7]

10 Esempi di output

Context 1 Tibet is a region on the Tibetan Plateau in Asia. It is the traditional homeland of the Tibetan people as well as some other ethnic groups such as Monpa, Qiang and Lhoba peoples and is now also inhabited by considerable numbers of Han Chinese and Hui people. Tibet is the highest region on Earth, with an average elevation of 4,900 metres (16,000 ft). The highest elevation in Tibet is Mount Everest, earth's highest mountain rising 8,848 m (29,029 ft) above sea level.

Answer 1: Mount Everest

- Ground Truth: What is the highest elevation in Tibet?
- Prediction: What is the highest point of the highest mountain point?

Answer 2: 16,000

- Ground Truth: What is the average elevation of Tibet, in feet?
- Prediction: What is the average population of the highest is?

Answer 3: 16,000 ft

- Prediction: What is the average high temperature in the in the?

Context 2 A revolution in 1332 resulted in a broad-based city government with participation of the guilds, and Strasbourg declared itself a free republic. The deadly bubonic plague of 1348 was followed on 14 February 1349 by one of the first and worst pogroms in pre-modern history: over a thousand Jews were publicly burnt to death, with the remainder of the Jewish population being expelled from the city. Until the end of the 18th century, Jews were forbidden to remain in town after 10 pm. The time to leave the city was signalled by a municipal herald blowing the Grüselhorn (see below, Museums, Musée historique);. A special tax, the Pflastergeld (pavement money), was furthermore to be paid for any horse that a Jew would ride or bring into the city while allowed to.

Answer 1: over a thousand

- Ground Truth: How many Jews were burned to death in 1349?
- Prediction: How many burned in the city?

Answer 2: special tax

- Ground Truth: What did the Jews need to pay to ride a horse into town?
- Prediction: What was the name of the r uss ian to convert to the?

Context 3 The Executive Board is responsible for the implementation of monetary policy (defined by the Governing Council) and the day-to-day running of the bank. It can issue decisions to national central banks and may also exercise powers delegated to it by the Governing Council . It is composed of the President of the Bank (currently Mario Draghi), the Vice-President (currently Vitor Constâncio) and four other members. They are all appointed for non-renewable terms

of eight years. They are appointed "from among persons of recognised standing and professional experience in monetary or banking matters by common accord of the governments of the Member States at the level of Heads of State or Government, on a recommendation from the Council, after it has consulted the European Parliament and the Governing Council of the ECB". The Executive Board normally meets every Tuesday.

Answer 1: Vitor Constâncio

- Ground Truth: Who is the Vice-President of The European Central Bank?
- Prediction: Who was the first president of the history?

Answer 2: sovereign debt

- Ground Truth: What must be a part of a states Gross Domestic Product in order for them to be considered for particpation in auctions?
- Prediction: What are one of the that can own safe from the institutions in the financial institutions?

Context 4 Strasbourg's status as a free city was revoked by the French Revolution. Enragés, most notoriously Eulogius Schneider, ruled the city with an increasingly iron hand. During this time, many churches and monasteries were either destroyed or severely damaged. The cathedral lost hundreds of its statues (later replaced by copies in the 19th century) and in April 1794, there was talk of tearing its spire down, on the grounds that it was against the principle of equality. The tower was saved, however, when in May of the same year citizens of Strasbourg crowned it with a giant tin Phrygian cap. This artifact was later kept in the historical collections of the city until it was destroyed by the Germans in 1870 during the Franco-Prussian war.

Answer 1: Eulogius Schneider

- Ground Truth: Who ruled the city with an iron hand?
- Prediction: What was the leader of the city in the city?

Answer 2: statues

- Ground Truth: What did the cathedrals lose in April 1794?
- Prediction: What did the most of the city in the?

Riferimenti bibliografici

[1] Papineni, Kishore, Salim Roukos, Todd Ward and Wei-Jing Zhu. "Bleu: a Method for Automatic Evaluation of Machine Translation." (2001)

[2] Kenji Imamura, Eiichiro Sumita. "Recycling a Pre-trained BERT Encoder for Neural Machine Translation." (2019)

[3] Dzmitry Bahdanau, Kyunghyun Cho, Yoshua Bengio. “Neural Machine Translation by Jointly Learning to Align and Translate.” (2015) “”

[4] Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, Yoshua Bengio. “Pointing the Unknown Words.” (2016)

[5] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser. “Attention is All You Need.” (2017)

[6] Sandeep Subramanian, Tong Wang, Xingdi Yuan, Saizheng Zhang, Yoshua Bengio, Adam Trischler. “Neural Models for Key Phrase Extraction and Question Generation.” (2018)

[7] Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever. “Language Models are Unsupervised Multitask Learners.” (2019)

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, Kristina Toutanova. “BERT: Pre-training of Deep Bidirectional Transformers forLanguage Understanding.” (2019)