Konkretnie o seq2seq algorytm istotny również teraz oczywiście zostało to wyparte przez transformery.

Algorytm seq2seq będziemy stosować tam gdzie jedną sekwencje chcemy zmienić w inną sekwencję np. tłumaczenie maszynowe czy podsumowywanie tekstów-> abstrakcyjne podsumowanie czyli opisanie tekstu za pomocą całkowicie innych słów a drugie to extracting summarization z tekstu wybieramy istotne fragmenty. Uczenie nadzorowane czyli potrzebujemy długie sekwencje tekstu i tego co oczekujemy np. odpowiadania na pytania.

Samo seq2seq-> jest to model oparty o architekturę enkoder dekoder. Sekwencje wejściową podajemy po kolei na wejście enkodera z tego obliczany jest stan ukryty a potem odtwarzany jest sygnał wejściowy.

Przykład

Mamy zdanie how are you ? uzyskujemy 4 tokeny które będziemy podawać po kolei na wejście a dalej do warstwy embednig która zamieni je na wektory i wchodzi to na koder który jest jakaś siecią rekurencyjną (RNN). Podajemy 1 koder wylicza nam sygnał h1 i wraca do kodera są przechowywane i tak dalej aż do h4 które przekazujemy na dekoder. Inicjować stan ukryty będziemy tym co wchodzi na dekoder czyli h4 jest to konkretny wektor całej sekwencji teraz w identyczny sposób będziemy przetwarzać w dekoderze też będzie warstwa embeding i na wejście będziemy podawać kolejne słowa przy czym zaczniemy od tokenu SOS podajemy je na dekoder który ma już jakiś stan ukryty przekazany z kodera. Dekoder wypluwa jakiś swój stan ukryty który do niego wraca oraz podawany jest na softmaxa który oblicza jakie słowo powinno być kolejne. Otrzymujemy sekwencję <SOS>, y1,y2… i teraz na wejście podawane jest ten yt z warstwy softmax i robimy to do momentu tokenu <EOS> czyli ten token musi być sztucznie dodany do słownika. Sygnał musimy generować to maksymalnej długości sekwencji najczęściej to średnia długość sekwencji ze zbioru uczącego.

GRU-> Gated recurat unit, jest to uproszczenie LSTM składa się z bramki aktualizacji i bramki restartu. Sygnał podawany jest na wejście stan ukryty z poprzedniej komórki idą razem sobie do bramek. Bramka restartu-> sigmoidalna macierz z wejściem plus macierz ze stanu ukrytego plus bias i znowu wartość między 0 a 1 tutaj nie do końca jak w bramce zapominania no ale można tak skojarzyć. Natomiast bramka aktualizacji to liczymy sygnał z wygląda analogicznie i to będzie pewnym balansem między stanem który możemy osiągnąć w danym kroku a tym co w poprzednim. Liczony jest taki stan który moglibyśmy uzyskać -> tangens hiperboliczny tym co wnosi podane słowo tym co mamy w stanie ukrytym iloczyn oczywiście plus bias. Stan jest akutalizowany w zależności od zt jeżeli blisko 0 to część sygnału się podmieni i tak dalej.

Zalety i wady w pdf.

Jeżeli mamy mniejszy zbiór danych to wybieramy GRU jeżeli zależność jest mała i LSTM by się przeuczał też warto użyć GRU. Jeżeli sekwencje są długie i początek ma znaczenie potem to bardziej użyjemy LSTM bo ma stan komórki.

GRU występuje w kilku wersjach Type 1, 2, 3 ograniczają GRU do pewnych rzeczy

Type 1-> sygnał zt jest uzależniony od stanu ukrytego i biasu niezależny od wejścia, ma to sens w zdaniach jeden do wielu

Type 2-> zakłada że stany będą uzależnianie tylko od poprzedniego stanu ukrytego nie uwzględniają biasu

Type 3-> uzależniony tylko od wyboru wyjść

BiLSTM-> bidarecional LSTM chodzi o to że sekwencja jest przetwarzana w obie strony są tworzone dwa niezależne LSTMy pierwszy generuje swój sygnał ukryty, a drugi generuje ten sygnał od końca. Stosujemy to raczej w koderze

Attention Mechanism mechanizm atencji-> w oryginalnym seq2seq tego nie było ale dodano później. Chodzi tutaj o to żeby wskazać który fragment tekstu ma znaczenie. Do dekodera był przekazywany jeden stan działanie kodera jest analogiczna cała sekwencja jest kodowana słowo po słowie i uzyskujemy stany ukryte a następnie przekażemy do dekodera cały zestaw stanów nie tylko jeden. Teraz dekoder ustala ważność tych stanów czyli oblicza współczynnik e t,i t to krok czasowy który z tych stanów ukryty jest ważniejszy. vT tanh(Whhi + Ws St-1) macierze wag uczone w ramach treningu podobnie jak v hi to wektor ukryty a st-1 to poprzedni stan. W efekcie otrzymamy wartość i stosujemy przybliżenie softmax na eti i mamy wektor o długości sekwencji wejściowej i tworzony jest wektor kontekstu ct obliczany jest jako suma istotności kolejnych stanów z wektora wejść średnia ważona i dopiero później stosowana jest LSTM na bazie poprzedniego stanu, słowa i wektora kontekstu i tak się oblicza stan ukryty przekazany na softmaxa który wygeneruj token.

Najczęściej stosuje się podejście że ostatni stan ukryty inicjalizuje dekoder. Najczęstsze podejście jest takie że wymiary są takie same ale nie jest to wymagane dodajemy warstwę FC i parametry są obliczane wsteczną propagacją błędów.

Uczenie seq2seq

Teacher Forcing-> to technika polegająca na tym jeśli mamy nasz dekoder to to co podajemy na wejście to nie poprzednio wygenerowana wartość tylko podajemy w procesie uczenia wartość taką jaka powinna tam być.

Embedding Layer-> inicjalizowane embding są losowe następnie się ucza i wydłuża to uczenie.

Seq2Seq Loss-> entropia krzyżowa softmax na tokenach i możemy sumować błąd

Sequence Padding-> uzupełnianie wartościami do konkretnej wartości albo 0 albo <PAD> pakiety mają funkcje wbudowane do tego

Wybór tokenu:

Greedy Decoding-> bazujemy na bieżącym stanie modelu i bierzemy kolejny kolejny

Samplin-> traktujemy wyjście softmaxa jako rozkład prawdopodobieństwa i losujemy największy

Top k SAmpling-> losujemy z k najlepszych

Beam Search-> polega na tworzeniu kilku niezależnych sekwencji wybieramy słowo i jego trzy warianty przechowywane są te najbardziej wiarygodne zależne od prawdopodobieństwa