-1

ترجمه‌ی ماشینی عصبی رویکردی است که اخیرا برای ترجمه‌ی ماشینی پیشنهاد شده‌است و بر خلاف سیستم‌ها ترجمه‌ی مبتنی بر عبارت که از تعداد بسیاری زیرمولفه تشکیل شده‌است که به صورت جداگانه تنظیم (tune) می‌شوند، ترجمه‌ی ماشینی عصبی سعی می‌کند یک شبکه‌ی عصبی بزرگ را بسازد و آموزش دهد که جمله‌ی ورودی را می‌خواند و ترجمه‌ی صحیح را برمی‌گرداند . اغلب مدل‌هایی که برای ترجمه‌ی ماشینی عصبی ارائه شده‌اند به خانواده‌ای از انکودر - دیکودر (encoder-decoder) تعلق دارند که جمله‌ی ورودی را به یک بردار با طول ثابت انکود کرده و سپس از روی این بردار دیکودر ترجمه را ایجاد می‌کند. سیستم انکودر - دیکودر ، که برای هر جفت زبان از انکودر و دیکودر تشکیل شده‌است که با هم آموزش می‌بینند تا احتمال ترجمه صحیح با توجه به جمله مبدأ را به حداکثر برسانند. گلوگاه این معماری در بهبود کارایی استفاده از یک بردار با طول ثابت است که ترجمه را برای جملات طولانی به خصوص جملاتی که از متون آموزش طولانی‌تر هستند را مشکل می‌کند. هر چه قدر جمله طولانی‌تر شود کارایی انکودر - دیکودر پایه به شدت بدتر می‌شود. برای حل این مشکل توسعه‌ی جدیدی از مدل انکودر - دیکودر معرفی می‌شود که در آن ترجمه و تراز کردن با هم صورت می‌گیرد. هر بار که مدل پیشنهادی یک کلمه برای ترجمه تولید می‌کند، جایگاه‌هایی از جمله‌ی مبدأ را جست‌وجو می‌کند (به صورت نرم) که مرتبط‌ترین اطلاعات در آن قرار دارد. پس از آن مدل کلمه‌ی هدف را بر اساس بردارهای زمینه‌ی مربوط این جایگاه‌ها و کلمات هدفی که پیش از این تولید شده‌اند، پیشبینی می‌کند. این به مدل اجازه می‌دهد تا بتواند با جمله‌های طولانی نیز کار کند.

معماری این مدل متشکل از RNN دو طرفه به عنوان انکودر و دیکودری است که جستجو در جمله اصلی را هنگام دیکود ترجمه شبیه سازی می‌کند.

احتمال شرطی کلمه هدف به شرط کلمات هدف قبلی و جمله‌ی ورودی به صورت زیر تعریف می‌شود:

$$p(y_i|y_1,\ldots,y_{i-1},\mathbf{x}) = g(y_{i-1}, s_i, c_i),$$

که s_i حالت پنهان (hidden state) در RNN در زمان i است و به صورت زیر محاسبه می‌شود:

$$s_i = f(s_{i-1}, y_{i-1}, c_i).$$

بر خلاف رویکردهای انکودر - دیکودر موجود در این معماری احتمال شرطی برای هر کلمه‌ی هدف مقدار متمایزی دارد.

بردار زمینه‌ی c_i به دنباله‌ای از annotationهای $(h_1, ..., h_{Tx})$ بستگی دارد که انکود جمله‌ی ورودی را به آن‌ها تصویر می‌کند. هر انوتیشن h_i شامل اطلاعاتی در مورد کل دنباله‌ی ورودی است اما تمرکز بیشتری بر قسمت‌هایی که اطراف کلمه i ام دنباله‌ی کلمات هستند دارد. بردار زمینه به صورت جمع وزن‌دار این انوتیشن‌ها محاسبه می‌شود.

$$c_i = \sum_{j=1}^{T_x} \alpha_{ij} h_j.$$

وزن α_{ij} برای هر انوتیشن h_j به صورت زیر محاسبه می‌شود:

$$\alpha_{ij} = \frac{\exp(e_{ij})}{\sum_{k=1}^{T_x} \exp(e_{ik})},$$

که $e_{ij} = a(s_{i-1}, h_j)$

یک مدل تراز بندی است که score می‌دهد ورودی های اطراف موقعیت j و خروجی در موقعیت i چقدر خوب تطبیق می‌یابند. این score بر اساس حالت مخفی s_{i-1} در RNN و j امین انوتیشن h_j روی دنباله‌ی ورودی است.

ما مدل a را به عنوان یک شبکه‌ی عصبی feed-forward که همراه با دیگر مولفه‌های سیستم آموزش می‌یابد پارامتر می‌کنیم.

می‌توان عمل جمع وزن‌دار روی انوتیشن‌ها را به صورت متوسط انوتیشن‌ها تفسیر کرد. که این متوسط‌گیری روی ترازهای ممکن صورت می‌گیرد. بگذارید α_{ij} احتمالی باشد که کلمه هدف y_i با یک کلمه مبدا x_j تراز شود یا از آن ترجمه شود. پس، بردار زمینه‌ی iام c_i متوسط حاشیه برای همه انوتیشن‌های با احتمال α_{ij} است.

این یک مکانیزم توجه را در دیکودر پیاده‌سازی می‌کند. دیکودر تصمیم می‌گیرد به چه قسمت‌هایی از جمله‌ی مبدا توجه کند. دیکودر با داشتن مکانیزم توجه ، انکودر را از مسئولیت انکود تمام اطلاعات در جمله مبدا در یک بردار با طول ثابت خلاص می‌کند.

RNN معمولی دنباله‌ی ورودی x را به ترتیب از اولین نماد x_1 تا آخرین x_{Tx} می‌خواند. با این حال ، در طرح پیشنهادی ، ما می خواهیم انوتیشن هر کلمه نه تنها کلمات قبلی ، بلکه کلمات بعدی را نیز خلاصه کند. بنابراین از RNN دو طرفه BiRNN استفاده کنیم. BiRNN شامل RNN های backward و forward است. RNNی که به صورت forward هست دنباله ورودی را به ترتیب (از x_1 تا x_{Tx}) می‌خواند و دنباله‌ی hidden stateهای

forward $(\overrightarrow{h}_1, \cdots, \overrightarrow{h}_{T_x})$. را محاسبه می‌کند.RNN ی که به صورت backward هست دنباله ورودی را برعکس (از x_{Tx} تا x_1) می‌خواند و دنباله‌ی hidden stateهای backward $(\overleftarrow{h}_1, \cdots, \overleftarrow{h}_{T_x})$. را محاسبه می‌کند. حال هر انوتیشن برای هر x_j را با اتصال hidden stateهای forward و backward آن به دست می‌آوریم.

$$h_j = \left[\overrightarrow{h}_j^\top; \overleftarrow{h}_j^\top\right]^\top$$