

راه اندازی یک مترجم ماشینی مبتنی بر شبکه عصبی

فراز یزدانی بنفشه درق، دکتر سیامک محمدی

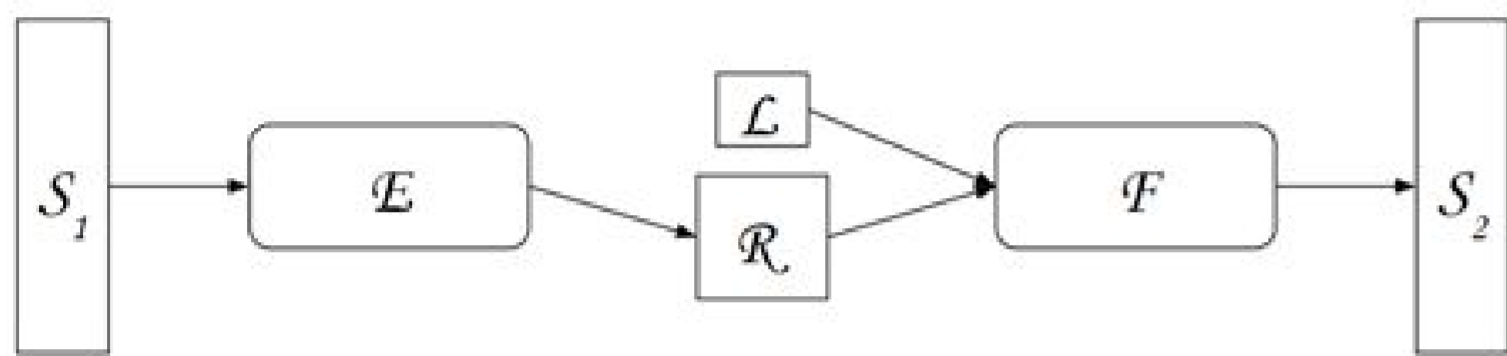
دانشکده مهندسی برق و کامپیوتر، دانشگاه تهران



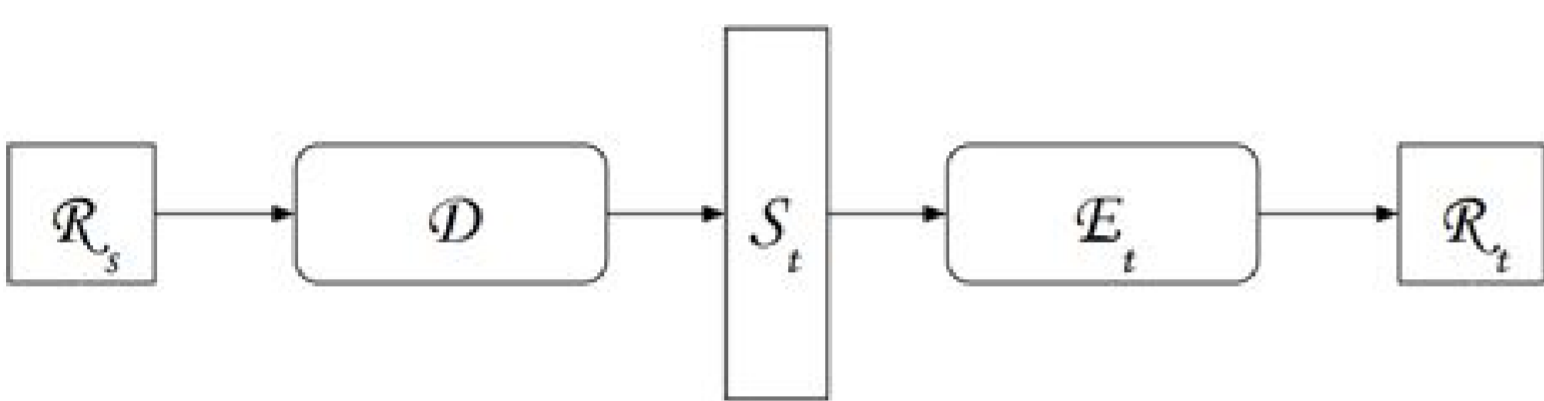
مقدمه

ترجمه‌ی ماشینی مبحثی قدیمی در بحث هوش مصنوعی و یادگیری ماشینی بوده و از ابتدای طرح آن تاکنون مراحل مختلفی برای آن داده شده است. در سال 2014، مقاله‌ای از Sutskever و همکاران [5] با معرفی مدل رمزگذار-رمزگشا با استفاده از LSTM انقلابی در این حوزه به وجود آورد. در همان سال، Bahdanau و همکاران با ارائه مکانیزم توجه [2]، توانستند کیفیت ترجمه را، به خصوص برای جمله‌های بلند، افزایش بدهند تا شبکه‌های عصبی عمیق به طور جدی به عنوان یک راحل کاربردی و در عین حال ساده وارد حوزه ترجمه ماشینی شود. در سال 2016، محققان گوگل با ارائه یک پیاده‌سازی عملی [6] و مقایسه پیکربندی‌های مختلف بر روی زبان‌های متفاوت، نقش مهمی در انتقال ترجمه ماشینی به وسیله شبکه‌های عصبی عمیق به صنعت داشتند. علی‌رغم نتایج خیره‌کننده‌ای که شبکه‌های عصبی عمیق در ترجمه ماشینی بدست آورده‌اند، اما محققان این حوزه همچنان با مشکلاتی مواجه هستند. به عنوان مثال، شبکه‌های عصبی عمیق به منظور یادگیری درست قوانین موردنظر نیازمند مقدار زیادی داده هستند. اما در بعضی زبان‌ها مانند فارسی با کمبود داده مواجه هستیم. در این پایان‌نامه برآن شدیم تا با ارائه چند تغییر روی چارچوب پیشفرض، این مشکلات را مرتفع کنیم. به همین منظور در طی دو فرآیند جداگانه و مستقل سعی شد تا با انجام تغییراتی در بعد نرم افزاری (نحوه ارائه داده به شبکه) و سخت افزاری (تغییر مسیرهای از پیش تعیین شده و معماری کلی شبکه) تاثیر این تغییرات بر روی پارامترهای مورد نظر سنجیده شود.

$$\text{Loss} = \text{crossEntropy}(S_s, S_t) + \text{leastSquare}(R_s, R_t)$$



شکل 1. معماری پیشنهادی برای آموزش رمزگذار



شکل 2. معماری پیشنهادی برای آموزش رمزگشا

نتایج

ایده‌های ذکر شده بر روی شبکه‌های دولایه با اندازه جاسازی 500، اندازه بردار LSTM برابر 1000 و اندازه کلمات 60 هزار تست شدن دو مقایسه کیفیت ترجمه ایده اول در مقایسه با حالت پایه نشان می‌دهد که پس از 20 دوره آموزش، دقت ترجمه مدل پایه در واحد BLEU برابر 37.91 و دقت ترجمه ایده برابر 37.35 است که به این معنی است که پیشرفتی وجود نداشته. برای ایده دوم تنها مرحله اولی پیاده سازی شد و به منزله یک مترجم مورد بررسی قرار گرفت. همچنین برای ایده دوم نیز ترجمه خروجی مدل از لحاظ معنایی و نحوی بسیار از حالت پایه فاصله داشت.

جمع بندی

همانطور که دیده شد، دو ایده پیشنهاد شده در افزایش کیفیت ترجمه ناتوان بودند. در مورد ایده اول این موضوع می‌تواند به دلیل عدم افزایش تعداد پارامترها در مقایسه با حالت پایه (علی‌رغم افزایش طول داده‌ها) بوده باشد. همچنین در مورد ایده دوم نیز ممکن است استفاده از مفهوم خودرمزگذار (زوج‌های فارسی-فارسی و انگلیسی-انگلیسی) برای شبکه درک منطق ترجمه مشکل ایجاد کرده باشد. مشکل دیگر می‌تواند این باشد که از بردار تک عضوی برای نشان دادن زبان خروجی نشان داده شده، در حالی که بردار one hot می‌تواند انتخاب بهتری باشد.

مراجع اصلی

1. R. Aharoni, Y. Goldberg, 2017, Towards string-to-tree neural machine translation. In Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics.
2. D. Bahdanau, K. Cho, Y. Bengio, 2014, Neural Machine Translation by Jointly Learning to Align and Translate, In <https://arxiv.org/abs/1409.0473>
3. M. Johnson, M. Schuster, Q. V. Le, M. Krikun, Y. Wu, Z. Chen, N. Thorat, F. Viégas, M. Wattenberg, G. Corrado, et al., 2017, Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation, In <https://arxiv.org/abs/1611.04558>
4. H. Schwenk, M. Douze, 2017, Learning Joint Multilingual Sentence Representations with Neural Machine Translation, In <https://arxiv.org/abs/1704.04154>
5. I. Sutskever, O. Vinyals, Q. V. Le, 2014, Sequence to sequence learning with neural networks. Advances in Neural Information Processing Systems 27 3104–3112.
6. Y. Wu, M. Schuster, Z. Chen, Q.V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, et al. 2016 Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, In <https://arxiv.org/abs/1609.08144>

مدل پیشنهادی

در ایده اول سعی بر آن شد تا با ارائه درخت پارس خطی‌شده به ورودی شبکه (با الهام از [1]) به بهبود کیفیت ترجمه شبکه کمک شود.

در ایده دوم با الهام از نحوه ترجمه انسان‌ها، و همچنین مقالات [3] و [4]، معماری شبکه را تغییر دادیم. در چارچوب پیشنهادی ما دو مرحله وجود دارد:

1. آموزش یک تابع که با گرفتن یک جمله در هر زبان دلخواه (موجود در داده‌ها)، بردار ادراک آن را تولید نماید، به گونه‌ای که بردار جملات مشابه در زبان‌های متفاوت، نزدیک به هم باشد. به این منظور از معماری مشابه شکل 1 استفاده می‌شود. زبان جملات S_1 و S_2 می‌تواند هرکدام از زبان‌های فارسی یا انگلیسی باشد (4 زوج ممکن) بردار تک‌عضوی L زبان S_2 را مشخص می‌کند.

2. آموزش یک تابع برای هر زبان مقصد، که با گرفتن یک بردار ادراک، جمله مربوط به آن در زبان مقصد را تولید می‌کند. به این منظور پس از آموزش رمزگذار در مرحله 1، از معماری شکل 2 استفاده می‌کنیم. ورودی این شبکه بردار ادراکی است که از ارائه جملات داده آموزش به رمزگذار E ایجاد می‌شوند. خروجی رمزگشای D را دوباره به E می‌دهیم و فاصله R_s و R_t را لحاظ می‌کنیم: