1 Introduction

Neural attention-based sequence-to-sequence models (seq2seq) (Sutskever et al., 2014; Bahdanau et al., 2014) have proven to be accurate and robust for many sequence prediction tasks. They have become the standard approach for automatic translation of text, at the cost of increased model complexity and uncertainty. End-to-end trained neural models act as a black box, which makes it difficult to examine model decisions and attribute errors to a specific part of a model. The highly connected and high-dimensional internal representations pose a challenge for analysis and visualization tools. The development of methods to understand seq2seq predictions is crucial for systems in production settings, as mistakes involving language are often very apparent to human readers. For instance, a widely publicized incident resulted from a translation system mistakenly translating “good morning” into “attack them” leading to a wrongful arrest (Hern, 2017).

In this work, we present the visual analysis tool SEQ2SEQ-VIS that allows interaction and “what if”-style exploration of trained seq2seq models through each stage of the translation process. The aim is to identify which patterns have been learned, to detect errors within a model, and to understand the model through counterfactual scenarios. In order to investigate the origin of an error within a seq2seq model, we separate errors within each translation stage into the following categories: (1) representation errors, in which an encoder or decoder misrepresent a word within a given context (2) alignment errors, in which the attention focuses on the wrong word, and (3) decoding errors, in which the prediction assigns a wrong probability distribution over words, or the beam search fails to include the correct solution.

We define three steps within an analysis that aim to understand the prediction process, understand how an output relates to training data, and examine causal relationships between inputs and outputs.

- **Examine Model Outputs:** SEQ2SEQ-VIS shows a separate visual representation for the output of each stage of the seq2seq pipeline.
- **Connect Outputs to Samples:** SEQ2SEQ-VIS connects the encoder and decoder of a seq2seq model to relevant training examples by showing a neighborhood of examples with the most similar internal states.
- **Test Alternative Decisions:** SEQ2SEQ-VIS enables “what if” explorations and causal relationship testing by manipulation of inputs, attention, and outputs.

The full system is shown in Figure 1. It combines visualizations for the external components with internal representations from specific examples and nearest-neighbor lookups over a corpus of precomputed examples. The entire system integrates with OpenNMT (Klein et al., 2017), one of the largest open source seq2seq libraries.

2 Debugging Use Case

This case study follows the example in Figure 1 and involves a model trainer (Strobelt et al., 2018b) who is building a German-to-English translation model on the IWSLT ’14 dataset (Mauro et al., 2012)). The user observes that a specific example was mistranslated. She finds the source sentence: Die längsten Reisen fangen an, wenn es auf den Straßen dunkel wird.

The correct translation for this sentence is The longest journeys begin, when it gets dark in the streets.

The model produces the mistranslation: The longest journey begins, when it gets to the streets. SEQ2SEQ-VIS shows the tokenized input sentence in blue and the corresponding translation of the model in yellow (on the top). The user observes that the model does not translate the word dunkel into dark. This mistake exemplifies several goals that motivated the development of
Seq2Seq-Vis. The user would like to examine the system’s decisions, connect to training examples, and test possible changes:

**Nearest neighbors to examine encoder and decoder:** Seq2Seq-Vis lets the user examine similar encoder/decoder states for any example. We define *neighborhood* as the twenty training examples with the closest states in vector space. Seq2Seq-Vis displays the neighborhood for a specific encoder state in a list of training set examples with red highlights for the word with the closest state. Figure 1(h) shows that the nearest neighbors for dunkel match similar use of the word from training data. Overall, the encoder seems to perform well in this case. A similar analysis can be done for the decoder.

**Graphical test for Attention error:** Another possible issue is that the attention may not focus on the corresponding source token dunkel. The previous test revealed many examples in the neighborhood that place dark after gets, which matches the valid translation. In Figure 1(c) the analyst can observe that the highlighted connection following get to the correct next word dunkel is very strong. Therefore, the user can assume that the attention is well set for predicting dark in this position.

**What-if testing for Prediction and Search error:** The combination of decoder state and attention is used to compute the probability over possible next words. It may be that an error occurs in this decision, leading to a poor probability of the word dark. The tool shows the most likely next words and their probabilities in Figure 1(d). Here, the model mistakenly assigns a higher probability to to than dark. However, both options are very close in probability, indicating that the model is almost equally split between the two choices. These local mistakes should be automatically fixed by the beam search, which is shown in Figure 1(e). In this case, the analyst finds that dark is never considered within the search. The analyst has identified a search error, where the approximations made by beam search cut off the better global option in favor of better local choices. To investigate whether the model would produce the correct answer the analyst can test a counterfactual, what would have happened if she had forced the translation to use dark at this critical position? By clicking on dark she can produce this probe (shown in Figure 1(i)), which yields the correct translation.

3 Conclusion

We have shown Seq2Seq-Vis, an interactive tool for finding errors in seq2seq models. It utilizes approaches to debugging in which black-box decisions are connected to easily understandable visual presentations. In future work, we will extend this work to improve models based on feedback from Seq2Seq-Vis. A longer description of the system, and additional use-cases can be found in Strobelt et al. (2018a).
References


Alex Hern. 2017. Facebook translates 'good morning' into 'attack them', leading to arrest. The Guardian.


