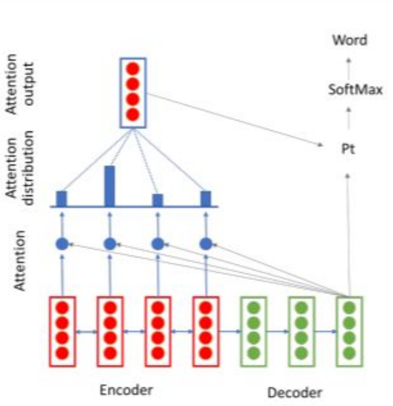
Improving RNN translation performance with attention

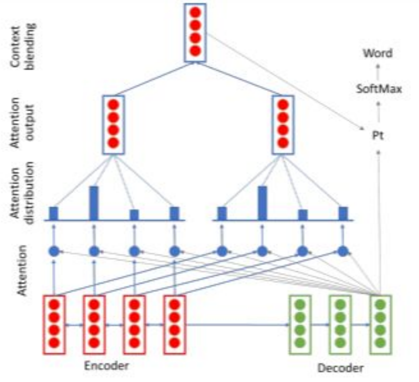
Patrick Kelly and Katie Fo

CS224N

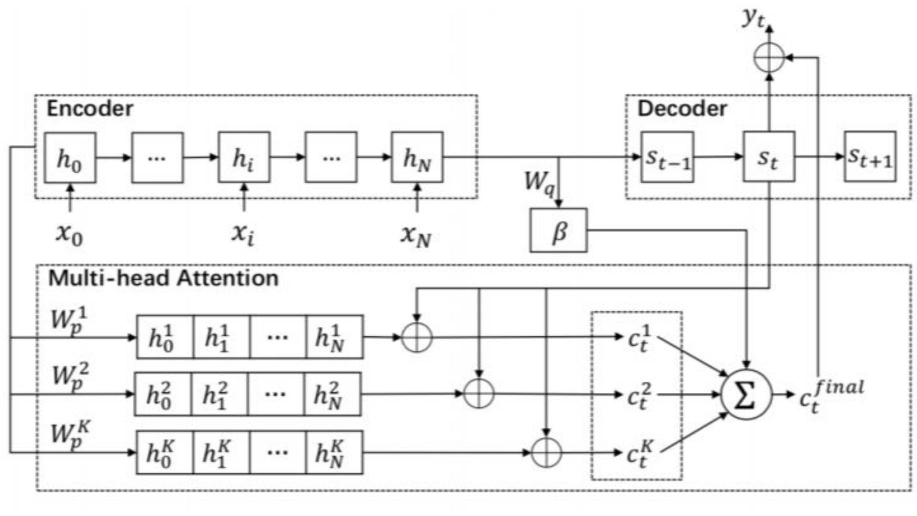
Basic, two-head, and multi-head attention models



(a) Basic Attention



(b) 2 Head attention - attention contexts were blended in two ways: concatenation with projection and additive



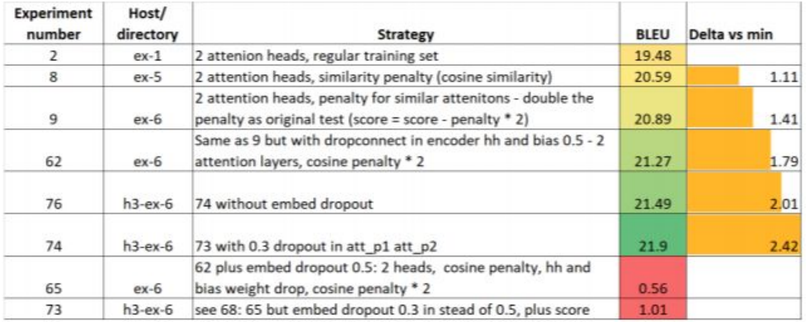
(c) Multihead attention as implemented by Tao et al. [5] uses additive attention. We implemented this approach with k=4. Figure taken from the original paper

Experimental setup

We used 4 Amazon Web Services p3.2xlarge instances, which house an NVIDIA P100 GPU, to run 76 distinct experiments. Experiments were run on an incremental basis, adding or varying features to test their impact on BLEU results. Data was the train/dev/test data with parallel Spanish/English sentences used in cs224n assignments. This data set consists of 216,617 train sentences, 851 dev sentences and 8064 test sentences. Each experiment has its own baseline and incremental impact on performance of each step is recorded.

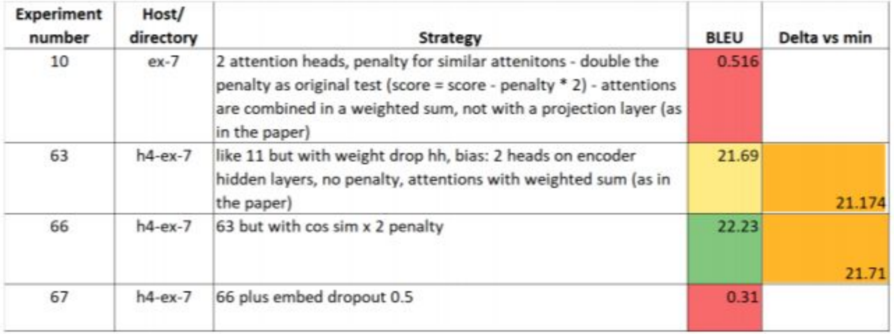
In total, three attention methods were tested: 2 head attention with projection, 2 head additive attention, and 4 head additive attention.

2 head attention with projection layers



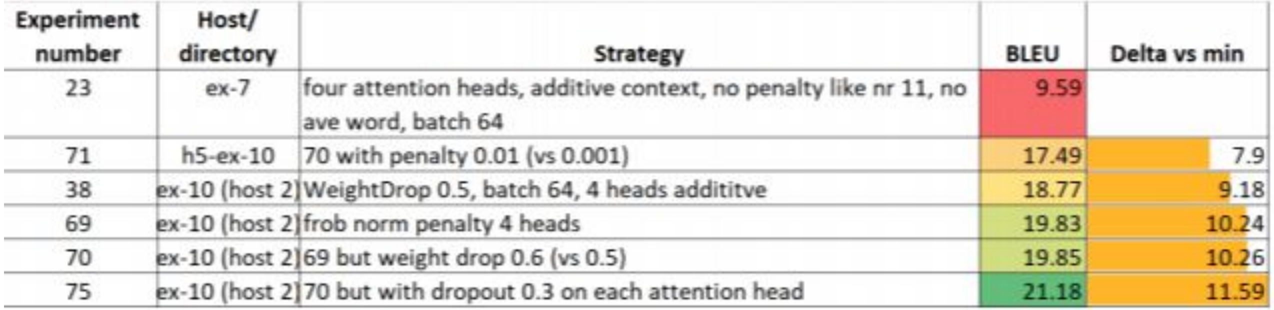
Experiment number	Host/ directory	Strategy	BLEU	Delta vs min
2	ex-1	2 attenion heads, regular training set	19.48	
8	ex-5	2 attention heads, similarity penalty (cosine similarity)	20.59	1.11
9	ex-6	2 attention heads, penalty for similar attenitons - double the penalty as original test (score = score - penalty * 2)	20.89	1.41
62	ex-6	Same as 9 but with dropconnect in encoder hh and bias 0.5 - 2 attention layers, cosine penalty * 2	21.27	1.79
76	h3-ex-6	74 without embed dropout	21.49	2.01
74	h3-ex-6	73 with 0.3 dropout in att_p1 att_p2	21.9	2.42
65	ex-6	62 plus embed dropout 0.5: 2 heads, cosine penalty, hh and bias weight drop, cosine penalty * 2	0.56	
73	h3-ex-6	see 68: 65 but embed dropout 0.3 in stead of 0.5, plus score	1.01	

2 head additive attention



Experiment number	Host/ directory	Strategy	BLEU	Delta vs min
10	ex-7	2 attention heads, penalty for similar attenitons - double the penalty as original test (score = score - penalty * 2) - attentions are combined in a weighted sum, not with a projection layer (as in the paper)	0.516	
63	h4-ex-7	like 11 but with weight drop hh, bias: 2 heads on encoder hidden layers, no penalty, attentions with weighted sum (as in the paper)	21.69	21.174
66	h4-ex-7	63 but with cos sim x 2 penalty	22.23	21.71
67	h4-ex-7	66 plus embed dropout 0.5	0.31	

4 head additive attention



Experiment number	Host/ directory	Strategy	BLEU	Delta vs min
23	ex-7	four attention heads, additive context, no penalty like nr 11, no ave word, batch 64	9.59	
71	h5-ex-10	70 with penalty 0.01 (vs 0.001)	17.49	7.9
38	ex-10 (host 2)	WeightDrop 0.5, batch 64, 4 heads additiтve	18.77	9.18
69	ex-10 (host 2)	frob norm penalty 4 heads	19.83	10.24
70	ex-10 (host 2)	69 but weight drop 0.6 (vs 0.5)	19.85	10.26
75	ex-10 (host 2)	70 but with dropout 0.3 on each attention head	21.18	11.59

Multi-head attention results

- Adding attention heads makes models more complex. Comprehensive analyses of architectures for hyperparameter tuning can involve hundreds of tests for a single architecture. Due to time and processing power (budget) limitations We only had the opportunity to test each architecture with 5 - 10 different variants. In this paper we tried to reach a general understanding of architectures and did not have time or resources to fine tune any particular model
- Models were very susceptible to gradient explosion - when adding a similarity penalty for 4 heads in particular - and vanishing gradients - when multihead architectures were not regularized with WeightDrop or dropouts.
- WeightDrop, consistent with the papers cited in this report, proved to be very effective in regularizing as stabilizing complex architectures.
- Embedding dropout caused vanishing gradients on 2 head models with additive and projected attention models. It, nevertheless provided an improvement in 4 head attention.

Learning punctuation

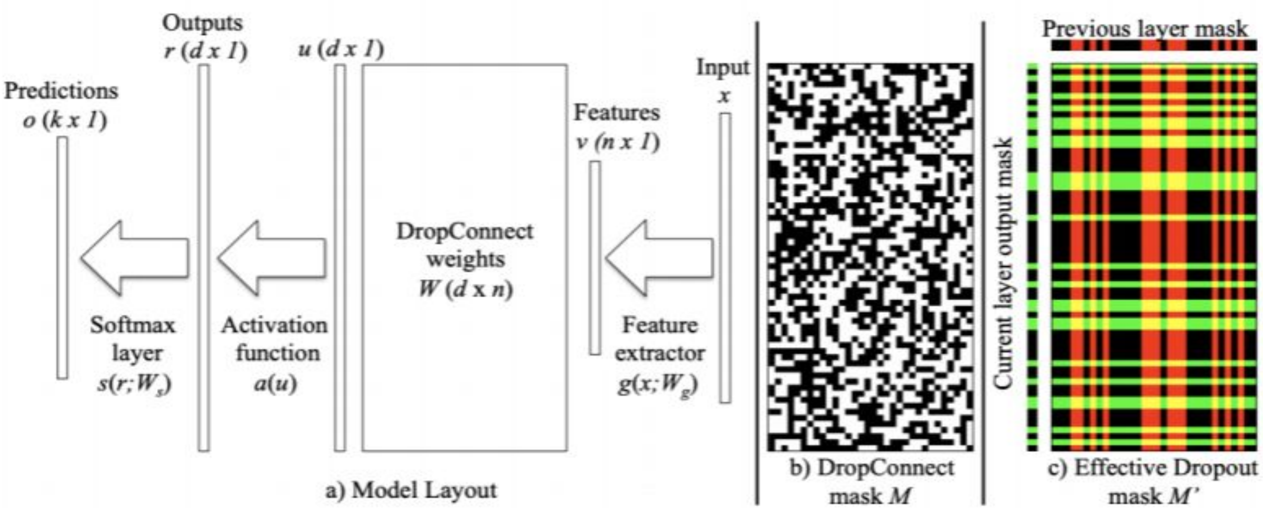
We developed training and test sets that separated commas and stops from their surrounding words. Training this way, BLEU scores increased dramatically, from a base of 22.60, to 32.00. To calculate the final score we glued the mentioned punctuation marks to their accompanying words to make comparisons with original results valid. The average increase in sentence BLEU was of 2.07.

Example: Correctly interpreting commas. Here the standard system inserts ’–’ where a comma is required. Our preprocessed system inserts the comma correctly.

- Source -> Creo que, como ella, a veces jugamos solos, y exploramos los lmites de nuestros mundos interior y exterior.
- Reference -> I think, like her, we sometimes play alone, and we explore the boundaries of our inner and our outer worlds.
- Standard, Sentence BLEU 20.09 -> I think, as we play – sometimes we play themselves, and we explore the limits of our inner worlds and outside.
- Preprocessed, Sentence BLEU 45.48 -> I think, like, we sometimes play alone, and we explore the limits of our inner worlds and outside.

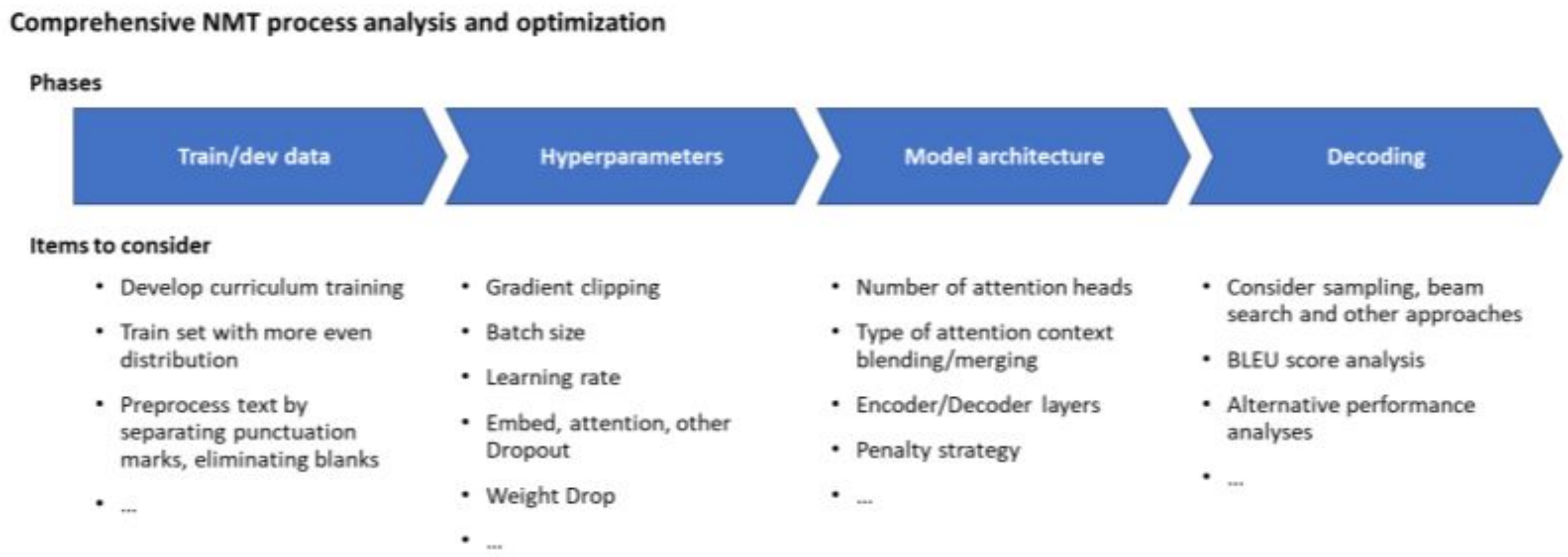
Regularization methods: DropConnect with WeightDrop

RNNs are especially hard to regularize since they tend to exhibit high variance and because hidden to hidden 3 connections should retain their continuity in order not to lose meaning.The weight drop approach generates a mask for all time-steps of the RNN, and thus there is consistent, regularized propagation of messages across time-steps. Masks change for each sentence. We tested the Weight Drop model to understand its effect on single and multihead attention models performance.



Improving NMT performance across the pipeline

Preprocessing has low cost and high impact, and should always be considered. Regularization, cost function penalties, and architecture decisions are more costly but necessary to fine tune the process.



Comprehensive NMT process analysis and optimization

Phases

Train/dev data → Hyperparameters → Model architecture → Decoding

Items to consider

Train/dev data:

- Develop curriculum training
- Train set with more even distribution
- Preprocess text by separating punctuation marks, eliminating blanks
- ...

Hyperparameters:

- Gradient clipping
- Batch size
- Learning rate
- Embed, attention, other Dropout
- Weight Drop
- ...

Model architecture:

- Number of attention heads
- Type of attention context blending/merging
- Encoder/Decoder layers
- Penalty strategy
- ...

Decoding:

- Consider sampling, beam search and other approaches
- BLEU score analysis
- Alternative performance analyses
- ...

Conclusion

In developing baseline 2 and 4 head architectures that can be used for further study and optimization. In exploring these architectures, we confirmed WeightDrop as one of the more effective techniques to improve system performance. We also found found that preprocessing by separating commas and full-stops form accompanying words raises BLEU performance substantially due to a better interpretation of punctuation and a lower incidence of OoV words. In summary:

- **WeightDrop in the encoder increases BLEU, especially in more complex systems with several attention heads, between 1 to 8 points. (This is not a formal benchmark.)**
- **Embedding dropout in some cases increased BLEU by approximately 0.5 points, but sometimes caused instability.**
- **Dropout on attention projections had a positive impact in the order of 0.5 BLEU points.**
- **Treating punctuation marks separately from words effectively improves word embeddings' connections to their meanings, punctuation marks embeddings more representative of their structural function in the sentence, and reduces OoV words.**