Improving Model Generalization Through "Theory of Mind" Ensembles

Kyle Eschen

Abstract:

Different word embedding schemes capture different semantic content, and thus have different strengths and weaknesses. In this work, I test whether one can enhance predictions made from pre-trained word embeddings by having models develop a "theory of mind" about the behavior of another model fed a different set of word embeddings. If a model can simultaneously predict its own classifications alongside alternative classifications of other models, it might be less prone to overfitting on the quirks of its own input. Using two models trained with different inputs (Universal Sentence Encoders and distilBert), I see if I can improve model performance on the Stanford Natural Language Inference task by having them learn each others output. I compare the performance of each model as a baseline to their "ToM enhanced" versions. Overall, this particular approach did not change either the quantity or the nature of the errors made by the models.

Introduction:

"Theory of mind" (ToM), or the ability to intuit the thoughts, feelings, and judgements of others, is a central component of human communication. In particular, the linguistic subfield of pragmatics studies the ways in which ToM is central to understanding connotation, implication, and context-sensitive meaning, domains that machine learning algorithms often struggle to navigate. This paper attempts to create a ToM-inspired ensemble, using word embeddings from the Universal Sentence Encoder and distilBert to improve performance on the Stanford Natural Language Inference task. I compare these "ToM-enhanced" models to a simple baseline ensemble consisting of the original models.

With humans, communication necessitates ToM between speaker and listener. Speakers must model how the listener will interpret their utterance (*first-order theory of mind*). However, because listeners interpret the utterance through their model of the listener's intention, speakers must also model how the listener models the speaker's intention (s*econd-order theory of mind*).

Such a framework can be extended to machine learning, as models can learn to "see the world through the eyes" of a model with different strengths and weaknesses . Imagine a two model ensemble with neural networks A and B. From A's perspective, we translate the concepts as follows:

Theory of Mind	**Social**	**Machine Learning**	**Loss is cross entropy**

Level	Interpretation	Interpretation	between model output and ___.
0th-Order	Seeing the world	The proper classification	Original labels
1st-Order	Seeing the world through someone else's eyes	Model B's labels	Model B's 0th-Order predictions
2nd-Order	Seeing oneself through someone else's eyes	Model B's prediction about model A's prediction	Model B's 1st-Order predictions

Unlike traditional ensemble methods, where different points of view about proper classification are produced in isolation and then aggregated, with a ToM approach each model attempts to develop an internal representation about how other models approach problems. The weights that inform zero-order predictions are colored by those that inform first and second-order predictions, and so there is a sense in which aggregation occurs upstream of the final pooling. From this angle, one can think about a ToM approach as an "internalized ensemble." By developing semi-distinct but overlapping representations of these different points of view, models may develop richer internal representations of the "semantic spread" of inputs, which in turn may help with generalization to new data.

Background:

There are several existing instances of ToM in the world of machine learning. Ensembles with tunable input models have an implicit element of ToM, as backpropagation forces models to adjust to become better "collaborators" with their fellow models. More directly, general adversarial networks (Goodfellow, 2014) involve one model attempting to fool another model, which requires that each model implicitly understands the behavior of the other.

Knowledge distillation (Hinton, 2015) provides a method for a smaller model (the student) to learn from a larger, more sophisticated model (the teacher). In such models, students simultaneously learn the proper classification (zero-order ToM), and their teacher's (first-order ToM). Significantly, the predictions of the teachers are "softened" by *temperature* (a proportional scaling down the logits that feed into the teacher softmax layer). By training on these fuzzy distributions in addition to the crisp point distributions of the training labels, students can better generalize to new data.

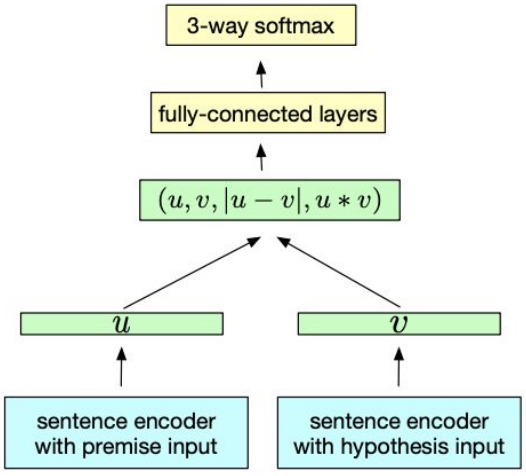
Finally, borne-again ensembles trains an initial model, copies it, and then trains the copy on both the training labels and it's previous predictions. A copy of the copy is made, and the process repeats. The outputs of all the copies are then fed into the a giant ensemble. Furlanello et al. (2018) reports that copies can outperform the original, which suggests that training on "second-order ToM" might be productive.

Methods (design and implementation)

I chose to ground the analysis in natural language inference, using Stanford Natural Language Inference (SNLI) data. SNLI provides two sentences, a hypothesis and a conclusion, and the model must determine whether the pair is a contradiction, entailment, or neutral.

To iterate quickly through different architectures to represent theory of mind, I chose to use pre-trained embeddings rather than fine tune deeper models. I decided to use embeddings from the universal sentence encoder and distilBert, as these embeddings were fixed length and known to preserve semantic integrity. USE and distilBert have different architectures (DAN and transformer), and are drawn from different corpuses, and were tuned on different tasks. This suggests that they likely capture different semantic aspects of sentences, and are thus plausible candidates for this experiment.

My baseline models (zero-order models) were classifiers built on top of each set of word embeddings, using the standard architecture to that described in Cer et al., 2018:



(Image credit to Cer et al., 2018)

Pre-trained embeddings for sentence 1 (the hypothesis) and sentence 2 (the premise) are concatenated alongside the difference between the two vectors and their Hadamard product,

which respectively capture analogical distance and similarity. The classifier layers consisted of two layers (one matching the input, and another with 128 nodes) that fed into a softmax classifier.

Zero-order models were trained against the training labels.

First-order models were initialized with the weights of their zero-order equivalents.I then added a "distillation arm" that was used to predict the outputs of other model from the previous round. (For instance, the USE model would have arm that predicted the true labels, and another that predicted the predictions of the zero-order distilBERT model.) This architecture was inspired by Hinton's above mentioned distillation paper. While I tried adjusting temperature on smaller test models, I found that any temperature than one led to a drop in performance. This was not surprising, as temperature models try to generalize the output of models with crisp, spiky distributions, and in this context we already had models with highly entropic output.

Second-order models were initialized from the their first-order analogues, and their distillation arm was replaced with a new one. These were trained both the training labels, and the predictions of the other model's first-order equivalent. (For instance, the second-order USE model predicts both the training labels and what the first-order distilBert model predicted the zero-order USE model predicted with regards to the training labels.)

Each model was trained for 20 epochs. The training data had 78734 rows, and the evaluation data had 9842.

Results and discussion

The baseline (zero-order) models had an accuracy on the evaluation set of 0.780 for USE and the 0.654 for BERT. I looked through the confusion matrices for both to see if either model seemed to make particular mistakes. The USE errors weird relatively uniform in their distribution. The biggest category of error for USE was to classify entailments as contradictions, but as I looked over corresponding sentences there was no clear patterns in length or content that might convey what tripped the model up. The BERT model tended to erroneously classify both contradictions and neutral statements as entailments.

For the first-order models, the USE model dipped marginally in accuracy on the evaluation set (0.779), while the BERT model improved (0.685). This is somewhat to be expected, as BERT was training using the distillations style on a stronger model, whereas USE was training on a weaker one. The changes to the confusion matrix for USE was minimal, and BERT had the exact same bias toward entailment classifications.

The second-order model saw marginal improvement for USE (0.781), and a drop for BERT (0.650). There was no discernible change in the confusion matrix for BERT or USE.

A complete table of accuracy, precision, recall, and F-Scores for the models across all rounds is supplied in the appendix.

(Unfortunately, I found that the model training histories did not save properly, so I was not able to recreate a graph of loss and accuracy over epochs in time. However, across all six models the accuracy and loss climbed / dipped and stabilized after around six or seven rounds, with little departure between training and evaluation. This suggests that the models were not overfitting.)

Conclusion

The ToM ensemble did nothing to improver classification accuracy while taking up considerably more computation time. The models also did not seem to change the kinds of errors made. The one improvement was that the the first-order BERT model seemed to learn from USE, as it saw a 3% bump in classification accuracy.

One possible cause of this disappointing result is that there seemed to be little issue with overfitting, which was one of the intended benefits of this approach. The training and evaluation loss seemed relatively coupled over the standardized 20 epochs of training. Another possible cause is that the word embeddings provided almost all of the benefit. Across all models, accuracy increased and flatlined after only five or six rounds, with few additional gains. This suggests that the word embeddings may have "plucked the low hanging fruit," and captured most of the easily acquirable semantic content.

Future explorations might include testing out this approach with the source models (USE and distilBert), as their richer structure may help with the deeper encoding of other models. Another approach might entail having models learn the outputs of a large number of other models, either individually or in aggregate. Yet another possible path for exploration would be to have encoders predict the standardized embeddings of other encoders. While experimenting, I tried to train USE to predict the distilBert embedding, but the loss flatlined after only a few epochs. However, developing this type ToM ensemble might still be worthwhile, as it could be used produce standardized embeddings of distilBert "through the eyes of USE", and possibly encode the strengths of both models.

Appendix - Tables:

Accuracy over ToM levels:

name	tom_level	accuracy
bert	0	0.654034
bert	1	0.685531
bert	2	0.650376
use	0	0.780634
use	1	0.779923
use	2	0.781650

Precision, recall, and F1-scores

name	tom_level	metric	entailment	neutral	contradiction
bert	0	precision	0.653887	0.709364	0.616469
bert	0	recall	0.737759	0.540958	0.680598
bert	0	f1-score	0.693296	0.613820	0.646948
bert	1	precision	0.632952	0.692515	0.768955
bert	1	recall	0.830880	0.634930	0.587858
bert	1	f1-score	0.718535	0.662474	0.666321
bert	2	precision	0.635778	0.594899	0.805755
bert	2	recall	0.770802	0.735394	0.444173
bert	2	f1-score	0.696809	0.657727	0.572665
use	0	precision	0.792076	0.757943	0.789553
use	0	recall	0.822770	0.715301	0.802318
use	0	f1-score	0.807131	0.736005	0.795884
use	1	precision	0.784533	0.724629	0.838443
use	1	recall	0.841093	0.754869	0.742526
use	1	f1-score	0.811830	0.739440	0.787575
use	2	precision	0.832320	0.735448	0.779251
use	2	recall	0.781316	0.738176	0.824893
use	2	f1-score	0.806012	0.736810	0.801423

Bibliography:

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St. John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, Yun-Hsuan Sung, Brian Strope, Ray Kurzweil: “Universal Sentence Encoder”, 2018;

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loic Barrault, Antoine Bordes: “Supervised Learning of Universal Sentence Representations from Natural Language Inference Data”, 2017; [http://arxiv.org/abs/1705.02364 arXiv:1705.02364]

Tommaso Furlanello, Zachary C. Lipton, Michael Tschannen, Laurent Itti, Anima Anandkumar: “Born Again Neural Networks”, 2018; [http://arxiv.org/abs/1805.04770 arXiv:1805.04770].

Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, Yoshua Bengio: “Generative Adversarial Networks”, 2014; [http://arxiv.org/abs/1406.2661 arXiv:1406.2661].

Geoffrey Hinton, Oriol Vinyals, Jeff Dean: “Distilling the Knowledge in a Neural Network”, 2015; [http://arxiv.org/abs/1503.02531 arXiv:1503.02531].