|  |
| --- |
| **Improving Upon REGMAPR** |

|  |  |
| --- | --- |
| **Dylan Sholtes**  Computer Science Department  Southern Illinois University Edwardsville  *dysholt@siue.edu* | **Christie McDaniel**  Computer Science Department  Southern Illinois University Edwardsville  *chmcdan@siue.edu* |
| **Tyler Norton**  Computer Science Department  Southern Illinois University Edwardsville  *tnorton@siue.edu* | **Ruslan Cahnovsky**  Computer Science Department  Southern Illinois University Edwardsville  *rcahnov@siue.edu* |

**1. Introduction**

While working with languages and sentences can come intuitively for humans, machines must be taught to understand the intricacies and relationships within language and between sentences. In particular, determining contradictions between claims can be natural for humans, but for machines there still lies room for improvement. Natural Language Processing (NLP) is still growing and has not yet caught up to the abilities of humans. Improving upon machines’ ability to identify contradictions within claims would allow for advances within fact-checking, the identification of fake news, analysis of text, and many other areas. Therefore our goal is to explore a novel approach to NLP and more specifically Natural language Inference (NLI) to improve upon machines’ ability to identify contradictions within pairs of sentences.

The input to our model will be pairs of sentences consisting of a text sentence and a hypothesis sentence. This input will be drawn from the Stanford Natural Language Inference (SNLI) Corpus, and the training data will also be accompanied by labels of either entailment, contradiction, or neutral. We will use the REGMAPR architecture to output a judgment on how the two sentences relate. Sentences with contradiction are labeled with “C,” those with entailment as “E,” and those that are neutral with “N.” The figure below shows our intended work-flow going forward.

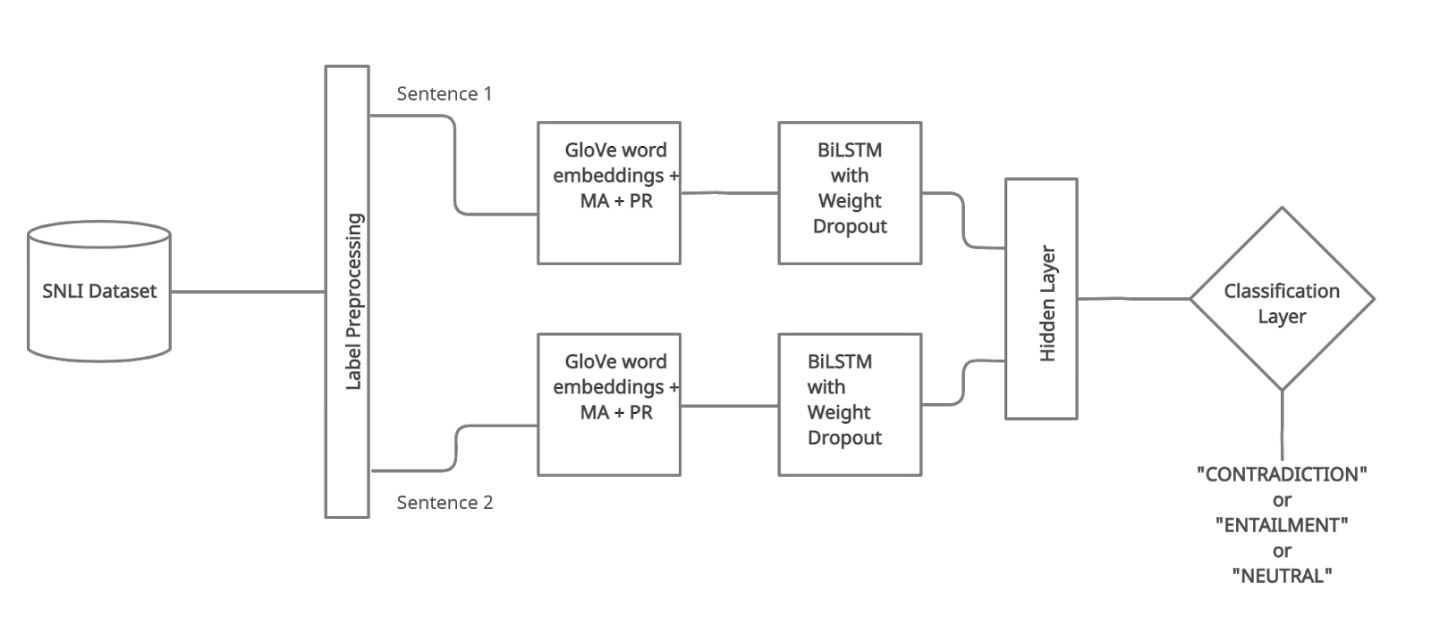


Figure 1: Shows an example of data preprocessing to training and making classifications.

**2. Related Work**

Several approaches exist for tackling NLP problems, among which REGMAPR takes on a more novel approach, first being used in 2018 [1]. REGMAPR varies from most established approaches by focusing on simplicity, such as how it applies a Siamese architecture. For established approaches, REGMAPR is most similar to the works of Wang and Jiang who use a compare-aggregate model to match sequences from sentence pairs [2]. REGMAPR shares similarity in approach since Wang and Jiang’s method relies on vectors and comparisons, which REGMAPR does as well though it uses a different structure. Whereas REGMAPR uses the Siamese architecture, Wang and Jiang’s use the compare-aggregate framework. This framework focuses on comparing vector representations of smaller units such as words, “and then aggregate these comparison results to make the final decision” [2]. He and Lin used a similar approach, also modeling pairwise word interactions and using a similarity focus layer [3]. Parikh et al. (2016) also used comparisons with words and attention-weighted versions to produce comparison vectors for use in aggregation [5]. All of the aforementioned models take an approach where they split up the data even further and apply vectors and comparisons to gain their results and high accuracy. Weaknesses of these models can be related to the comparison function used, since enough attention is not always paid to it. Additionally, the extent to which these approaches can be applied has not fully been explored since they haven’t been widely tested [2].

Differing approaches include NLP models that use a BiLSTM with intersentence attention. Examples include a proposed model by Liu et al. (2016), which uses BiLSTM with a sentence encoding module to extract relationships between sentences [4]. Other sentence encoding-based models include the, “LSTMs-based model, GRUs-based model, TBCNN-based model and SPINN-based model” [4]. However, these suffered from not fully utilizing contextual information, which Liu et al. improved upon in their model. Similarly to REGMAPR, however, is that Liu et al.’s model employs a Siamese architecture where it has two identical sentence encoders. Mueller and Thyagarajan (2016) made use of the aforementioned LSTM-based model, adapting it for use in a Siamese architecture [10]. In their results, they showed how a simple LSTM model could be trained with enough data to, “learn a highly structured space of sentence-representations that captures rich semantics,” while sidestepping complexity of prior LSTM and other models [10].

From the prior work and models, it would seem that approaches that highlight simplicity are currently gaining traction and proving to be just as successful, if not more so, than more complex models. In particular, this is seen in the increased use of the Siamese architecture being applied to NLP problems where it had not been before. Utilizing a Siamese architecture is becoming more state-of-the-art as it proves itself to be more effective. This can even be seen in REGMAPR’s adaptation of the architecture which allowed it to remain much simpler in implementation while producing highly accurate results as seen in those published in the SNLI Corpus [6]. While the task of classifying sentences can be performed by hand and was performed this way for the creation of the datasets, for vast amounts of data or complex sentence pairs, models such as the ones mentioned prove to be very efficient and thus can save time from a sometimes menial task. Additionally, advances in automating the task of sentence classification could allow for more instant analysis of text which could prove beneficial in many different fields.

**3. Materials and Method**

**3.1 Dataset and Features**

The primary dataset we will use for training comes from the SNLI Corpus and includes three text files: a dev file containing 10,000 sentence pairs, a training file containing 550,152 pairs, and a test file containing 10,000 pairs [6]. The sentence pairs are human-written in English and were manually labeled for balanced classification.  The SNLI dataset includes fields for the pairs of sentences, two different parses from the Stanford Parser, annotator and gold labels for classification accuracy, and a unique ID for each pair of sentences.

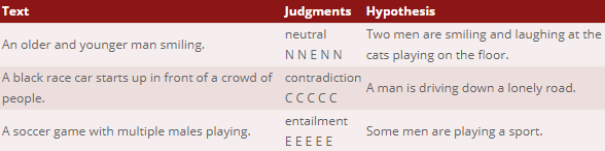


Figure 2: Shows examples from the SNLI dataset.

The Paraphrase Detection Database version 2.0 (PPDB) made by Pavlick et al. (2015) will also be utilized within the paraphrase detection layer of the REGMAPR model [7]. PPDB contains over 100 million paraphrases in 16 different languages. Additionally, the second version of this dataset contains improvements in many areas including scoring for ranking paraphrases as well as entailment relation in each pair (seen in the symbols in Figure 3).

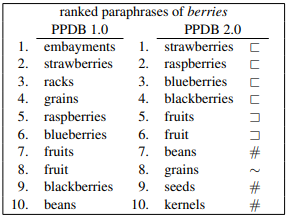


Figure 3: Examples of paraphrases from PPDB version 2.0.

Additionally, the Sentences Involving Compositional Knowledge (SICK) dataset was used to test the baseline model and similar models [9]. This dataset contains nearly 10,000 pairs of sentences labeled for entailment relation as well as annotated for relatedness in meaning shown as a gold score on a 5-point rating scale.

For testing the baseline model with the SICK dataset, data was preprocessed by converting the semantic relatedness score from a scale of 1 to 5 down to a scale of 0 to 1 for the classification layer. This is done through a simple python script that uses pandas to read the dataset and convert the scale down, and returns the dataset as a tsv. For the SNLI dataset, preprocessing will have to occur to convert the textual entailment label to a binary label used for the classification layer.

REGMAPR makes use of two features, a matching (MA) feature, as well as a paraphrase detection (PR) feature. The matching feature detects an exact word match between the pair of sentences, while the paraphrase detection feature makes use of the PPDB to detect paraphrases between the sentence pairs. These features add an additional two layers to the word embeddings.

**3.2 Methods**

REGMAPR approaches NLP with a Siamese architecture, which makes two or more identical subnetworks that share weights and can compare feature vectors. The input sentences are then encoded, “using a BiLSTM and the representations are composed by computing the element-wise absolute difference and product” [1]. Further, the embeddings of words are augmented with a matching feature to determine if the same words are shared between sentence pairs. REGMAPR utilizes an external database of semantically related words to identify semantic dependence without being too constricted by the aforementioned embedding; this adds a paraphrase feature to the embedding of words, resulting in two additional dimensions for the embeddings that capture syntactic and semantic interaction between the sentence pairs. REGMAPR is defined by its base Siamese architecture and two dimensions of embeddings on words. However, to avoid overfitting regularization is applied; for REGMAPR specifically, four types of regularization were used to train the models and then compare the results.

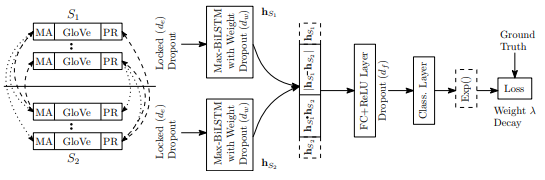


Figure 4: Schematic diagram of REGMAPR model from Brahma (2018) [1].

More specifically, REGMAPR works by encoding each sentence into a vector using BiLSTM, then max-pooling is used on the intermediate states produced by BiLSTM. Feature vectors are then made via the encodings of the sentences which are, “composed by concatenating the element-wise absolute difference and element-wise product with the original vectors” [1]. The feature vector is passed through a fully connected layer and then ReLU activation, and finally a classification or scoring layer.

REGMAPR also makes use of four types of regularization. Locked Dropout is applied after the embedding layer, Simple Dropout is applied after the ReLU activation, Weight Dropout is applied to recurring weights in the BiLSTM encoder, and L2 Weight Decay regularization is applied for the loss.

Combining the matching (MA) and paraphrase (PR) features results in an equation shown in figure 5. It uses a 300 dimensional GloVe, where GloVe is an unsupervised learning algorithm to obtain vector representations of words where *t* represents a word in the sentence [8]. The set of words are denoted by T(S3-i) where i ∈ {1, 2}, with the first sentence being represented by S1, and the second sentence being represented by S2. The purpose of using T(S3-i) is for when computing ESi(*t*), the word *t* is checked against the other sentence in the pair.  𝟙 represents the indicator function, stating whether the word has an exact match in the set of words T(S3-i) in the matching feature, or if T(S3-i) contains a paraphrase match in the set P(*t*) in the paraphrase feature. P(*t*) represents the set where P(*t*) = {*t’* | *t’* is a paraphrase in the PPDB}[1]. Both the matching and paraphrase features are binary features, giving results of 0 or 1.

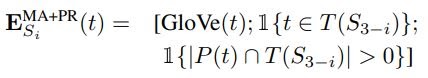


Figure 5: Equation for the combination of matching and paraphrase features.

For our hyperparameters, we used an embedding dimension size of 300 and optimized using Adam with a learning rate of 0.001. For regularization, hyperparameters of 0.4 for Locked Dropout, 0.4 for Simple Dropout, and 0.2 for the Recurrent Weight Dropout. The model is trained over 10 epochs, with batch sizes of 64 for the training set, and batch sizes of 128 for the validation set. These hyperparameters were chosen as they were used by Brahma (2018) who originally proposed the REGMAPR model and achieved good results [1]. This gives our group a good baseline for comparison against Brahma’s model, and allows us to tune these hyperparameters to increase the accuracy.

**4. Preliminary Results and Next Steps**

So far, our group has run our baseline REGMAPR model on the SICK dataset. The SICK dataset is similar to the SNLI dataset; however, sentence pairs are given a semantic relatedness score, instead of a textual entailment label. We split the SICK dataset into a training and validation set, using 8000 sentences for training, and 2000 sentences for the validation set. In doing this, we achieved minimal loss in the semantic relatedness score. Our next step is to convert our baseline model to use the SNLI dataset, and test for textual entailment instead of relatedness score. In doing this, we will have to change the loss function used, as REGMAPR uses a mean squared error loss for semantic relatedness, while it uses a cross entropy loss function for textual entailment. We will also have to preprocess our data by binarizing the textual entailment labels, to allow the classification layer to give predictions on textual entailment. Once we have done the preprocessing to allow our model to use the SNLI dataset, our goal will be to run experiments by changing parameters and dimensions, to achieve higher than the recorded 85.8% accuracy on the SNLI Corpus. We plan on changing the dimensions of the GloVe word embeddings as well as the LSTM hidden dimension. Other experiments we plan on running include changing the regularization hyperparameters, as well as the learning rate for the optimizer. We also plan on trying out other optimizers, such as Nadam and Adamax. Both Nadam and Adamax are improvements upon Adam, possibly yielding us a better accuracy.

**References**

[1] Brahma, S. (2018). REGMAPR-Text Matching Made Easy. *arXiv preprint arXiv:1808.04343*.

<https://arxiv.org/pdf/1808.04343v1.pdf>

[2] Wang, S., & Jiang, J. (2016). A compare-aggregate model for matching text sequences. *arXiv preprint arXiv:1611.01747*.

<https://arxiv.org/pdf/1611.01747.pdf>

[3] He, H., & Lin, J. (2016, June). Pairwise word interaction modeling with deep neural networks for semantic similarity measurement. In *Proceedings of the 2016 conference of the north American chapter of the Association for Computational Linguistics: human language technologies* (pp. 937-948).

<https://www.aclweb.org/anthology/N16-1108.pdf>

[4] Liu, Y., Sun, C., Lin, L., & Wang, X. (2016). Learning natural language inference using bidirectional LSTM model and inner-attention. *arXiv preprint arXiv:1605.09090*.

<https://arxiv.org/pdf/1605.09090v1.pdf>

[5] Parikh, A. P., Täckström, O., Das, D., & Uszkoreit, J. (2016). A decomposable attention model for natural language inference. *arXiv preprint arXiv:1606.01933*.

<https://arxiv.org/pdf/1606.01933.pdf>

[6] Bowman, S. R., Angeli, G., Potts, C., & Manning, C. D. (2015). A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

<https://nlp.stanford.edu/projects/snli/>

[7] Pavlick, E., Rastogi, P., Ganitkevitch, J., Van Durme, B., & Callison-Burch, C. (2015, July). PPDB 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)* (pp. 425-430).

<https://www.aclweb.org/anthology/P15-2070.pdf>

[8] Pennington, J., Socher, R., & Manning, C. D. (2014, October). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)* (pp. 1532-1543).

<https://www.aclweb.org/anthology/D14-1162.pdf>

[9] Marelli, M., Menini, S., Baroni, M., Bentivogli, L., Bernardi, R., & Zamparelli, R. (2014, May). A SICK cure for the evaluation of compositional distributional semantic models. In *Lrec* (pp. 216-223).

<http://www.lrec-conf.org/proceedings/lrec2014/pdf/363_Paper.pdf>

[10] Mueller, J., & Thyagarajan, A. (2016, March). Siamese recurrent architectures for learning sentence similarity. In *Proceedings of the AAAI Conference on Artificial Intelligence* (Vol. 30, No. 1).

<https://people.csail.mit.edu/jonasmueller/info/MuellerThyagarajan_AAAI16.pdf>