Element-wise Bilinear Interaction for Sentence Matching

Jihun Choi, Taeuk Kim, Sang-goo Lee
Department of Computer Science and Engineering
Seoul National University, Seoul, Korea
{jhchoi,taeuk,sglee}@europa.snu.ac.kr

Abstract

When we build a neural network model predicting the relationship between two sentences, the most general and intuitive approach is to use a Siamese architecture, where the sentence vectors obtained from a shared encoder is given as input to a classifier. For the classifier to work effectively, it is important to extract appropriate features from the two vectors and feed them as input. There exist several previous works that suggest heuristic-based function for matching sentence vectors, however it cannot be said that the heuristics tailored for a specific task generalize to other tasks. In this work, we propose a new matching function, *ElBiS*, that learns to model element-wise interaction between two vectors. From experiments, we empirically demonstrate that the proposed ElBiS matching function outperforms the concatenation-based or heuristic-based matching functions on natural language inference and paraphrase identification, while maintaining the fused representation compact.

1 Introduction

Identifying the relationship between two sentences is a key component for various natural language processing tasks such as paraphrase identification, semantic relatedness prediction, textual entailment recognition, etc. The most general and intuitive approach to these problems would be to encode each sentence using a sentence encoder network and feed the encoded vectors to a classifier network.[1]

For a model to predict the relationship correctly, it is important for the input to the classifier to contain appropriate information. The most naïve method is to concatenate the two vectors and delegate the role of extracting features to subsequent network components. However, despite the theoretical fact that even a single-hidden layer feedforward network can approximate any arbitrary function (Cybenko, 1989; Hornik, 1991), the space of network parameters is too large, and it is helpful to narrow down the search space by directly giving information about interaction to the classifier model, as empirically proven in previous works built for various tasks (Ji and Eisenstein, 2013; Mou et al., 2016; Xiong et al., 2016, to name but a few).

In this paper, we propose a matching function which learns from data to fuse two sentence vectors and extract useful features. Unlike bilinear pooling methods designed for matching vectors from heterogeneous domain (e.g. image and text), our proposed method utilizes element-wise bilinear interaction between vectors rather than interdimensional interaction. In §3, we will describe the intuition and assumption behind the restriction of interaction.

This paper is organized as follows. In §2, we briefly introduce previous work related to our objective. The detailed explanation of the proposed model is given in §3, and we show its effectiveness in extracting compact yet powerful features in §4. §5 concludes the paper.

2 Related Work

As stated above, matching sentences is a common component in various tasks in natural language processing. Ji and Eisenstein (2013) empirically prove that the use of element-wise multiplication and absolute difference as matching function substantially improve performance on paraphrase identification, and Tai et al. (2015) apply the same matching scheme to the semantic related-

[1] The encoded vectors can also be fed into a regression network, however in this work we focus only on classification.

*Proceedings of the 7th Joint Conference on Lexical and Computational Semantics (*SEM)*, pages 107–112
New Orleans, June 5-6, 2018.

ness prediction task. Mou et al. (2016) show that using the element-wise multiplication and difference along with the concatenation of sentence vectors yields good performance in natural language inference, despite redundant components such as concatenation and element-wise difference. Yogatama et al. (2017) and Chen et al. (2017) use modified versions of the heuristics proposed by Mou et al. (2016) in natural language inference.

However, to the best of our knowledge, there exists little work on a method that adaptively learns to extract features from two sentence vectors encoded by a shared encoder. Though not directly related to our work's focus, there exist approaches to fuse vectors from a homogeneous space using exact or approximate bilinear form (Socher et al., 2013; Lin et al., 2015; Wu et al., 2016; Krause et al., 2016).

There have been several works for extracting features from two heterogeneous vectors. Wu et al. (2013) use a bilinear model to match queries and documents from different domains. Also, approximate bilinear matching techniques such as multimodal compact bilinear pooling (MCB; Fukui et al., 2016), low-rank bilinear pooling (MLB; Kim et al., 2017), and factorized bilinear pooling (MFB; Yu et al., 2017) are successfully applied in visual question answering (VQA) tasks, outperforming heuristic feature functions (Xiong et al., 2016; Agrawal et al., 2017).

MCB approximate the full bilinear matching using Count Sketch (Charikar et al., 2002) algorithm, MLB and MFB decompose a third-order tensor into multiple weight matrices, and MUTAN (Ben-younes et al., 2017) use Tucker decomposition to parameterize bilinear interactions. Although these bilinear pooling methods give significant performance improvement in the context of VQA, we found that they do not help matching sentences encoded by a shared encoder.

3 Proposed Method: *ElBiS*

As pointed out by previous works on sentence matching (Ji and Eisenstein, 2013; Mou et al., 2016), heuristic matching functions bring substantial gain in performance over the simple concatenation of sentence vectors. However, we believe that there could be other important interaction that simple heuristics miss, and the optimal heuristic could differ from task to task. In this section, we propose a general matching function that learns to extract compact and effective features from data.

Let $\mathbf{a} = (a_1, \cdots, a_d) \in \mathbb{R}^d$ and $\mathbf{b} = (b_1, \cdots, b_d) \in \mathbb{R}^d$ be sentence vectors obtained from a encoder network.[2] And let us define $\mathbf{G} \in \mathbb{R}^{d \times 3}$ as a matrix constructed by stacking three vectors $\mathbf{a}, \mathbf{b}, \vec{\mathbf{1}} \in \mathbb{R}^d$ where $\vec{\mathbf{1}}$ is the vector of all ones, and denote the i-th row of $\mathbf{G}$ by $\mathbf{g}_i$.

Then the result of applying our proposed matching function, $\mathbf{r} = (r_1, \cdots, r_d) \in \mathbb{R}^d$, is defined by

$$r_i = \phi\left(\mathbf{g}_i^\top \mathbf{W}_i \mathbf{g}_i\right), \qquad (1)$$

where $\mathbf{W}_i \in \mathbb{R}^{3\times 3}, i \in \{1, \cdots, d\}$ is a matrix of trainable parameters and $\phi(\cdot)$ an activation function ($\tanh$ in our experiments).

Due to its use of bilinear form, it can model every quadratic relation between a_i and b_i, i.e. can represent every linear combination of $\{a_i^2, b_i^2, a_i b_i, a_i, b_i, 1\}$. This means that the proposed method is able to express frequently used element-wise heuristics such as element-wise sum, multiplication, subtraction, etc., in addition to other possible relations.[3]

Further, to consider multiple types of element-wise interaction, we use a set of M weight matrices per dimension. That is, for each $\mathbf{g}_i$, we get M scalar outputs $(r_i^1, \cdots, r_i^M)$ by applying Eq. 1 using a set of separate weight matrices $(\mathbf{W}_i^1, \cdots, \mathbf{W}_i^M)$:

$$r_i^m = \phi\left(\mathbf{g}_i^\top \mathbf{W}_i^m \mathbf{g}_i\right). \qquad (2)$$

Implementation-wise, we vertically stack G for M times to construct $\tilde{\mathbf{G}} \in \mathbb{R}^{Md\times 3}$, and use each row $\tilde{\mathbf{g}}_i$ as input to Eq. 1. As a result, the resulting output $\mathbf{r}$ becomes a Md-dimensional vector:

$$r_i = \phi\left(\tilde{\mathbf{g}}_i^\top \mathbf{W}_i \tilde{\mathbf{g}}_i\right), \qquad (3)$$

where $\mathbf{W}_i \in \mathbb{R}^{3\times 3}, i \in \{1, \cdots, Md\}$. Eq. 1 is the special case of Eq. 2 and 3 where $M = 1$. We call our proposed element-wise bilinear matching function *ElBiS* (Element-wise Bilinear Sentence Matching).

Note that our element-wise matching requires only $M \times 3 \times 3 \times d$ parameters, the number of

[2] Throughout this paper, we assume a d-dimensional vector is equivalent to the corresponding $d \times 1$ matrix.

[3] Though a bilinear form cannot represent the absolute difference between inputs, note that $(a_i - b_i)^2 = a_i^2 - 2a_i b_i + b_i^2$ can alternatively represent commutative difference. Yogatama et al. (2017) use this quadratic form instead of the absolute difference.

which is substantially less than that of full bilinear matching, Md^3. For example, in the case of $d = 300$ and $Md = 1200$ (the frequently used set of hyperparameters in NLI), the full bilinear matching needs 108 million parameters, while the element-wise matching needs only 10,800 parameters.

Why element-wise? In the scenario we are focusing on, sentence vectors are computed from a Siamese network, and thus it can be said that the vectors are in the same semantic space. Therefore, the effect of considering interdimensional interaction is less significant than that of multimodal pooling (e.g. matching a text and a image vector), so we decided to model more powerful interaction within the same dimension instead. We also would like to remark that our preliminary experiments, where MFB (Yu et al., 2017) or MLB (Kim et al., 2017) was adopted as matching function, were not successful.

4 Experiments

We evalute our proposed ElBiS model on the natural language inference and paraphrase identification task. Implementation for experiments will be made public.

4.1 Natural Language Inference

Natural language inference (NLI), also called recognizing textual entailment (RTE), is a task whose objective is to predict the relationship between a premise and a hypothesis sentence. We conduct experiments using Stanford Natural Language Inference Corpus (SNLI; Bowman et al., 2015), one of the most famous dataset for the NLI task. The SNLI dataset consists of roughly 570k premise-hypothesis pairs, each of which is annotated with a label (entailment, contradiction, or neutral).

For sentence encoder, we choose the encoder based on long short-term memory (LSTM; Hochreiter and Schmidhuber, 1997) architecture as baseline model, which is similar to that of Bowman et al. (2015) and Bowman et al. (2016). It consists of a single layer unidirectional LSTM network that reads a sentence from left to right, and the last hidden state is used as the sentence vector. We also conduct experiments using a more elaborated encoder model, Gumbel Tree-LSTM (Choi et al., 2018). As a classifier network, we use an MLP with a single hidden layer. In experiments with heuristic matching we use the heuristic features proposed by Mou et al. (2016) and adopted in many works on the NLI task: $[\mathbf{a}; \mathbf{b}; \mathbf{a}-\mathbf{b}; \mathbf{a}\odot\mathbf{b}]$, where $\mathbf{a}$ and $\mathbf{b}$ are encoded sentence vectors. For more detailed experimental settings, we refer readers to §A.1.

Matching Fn.	**# Params.**	**Acc. (%)**
Concat	1.34M	81.6
Heuristic	1.96M	83.9
ElBiS ($M = 1$)	1.04M	84.4
ElBiS ($M = 2$)	1.35M	84.5
ElBiS ($M = 3$)	1.66M	**85.0**
ElBiS ($M = 4$)	1.97M	84.6

Table 1: Results on the SNLI task using LSTM-based sentence encoders.

Matching Fn.	**# Params.**	**Acc. (%)**
Concat	2.25M	82.4
Heuristic	2.86M	84.6
ElBiS ($M = 1$)	1.94M	84.8
ElBiS ($M = 2$)	2.25M	85.6
ElBiS ($M = 3$)	2.56M	**85.9**
ElBiS ($M = 4$)	2.87M	85.6

Table 2: Results on the SNLI task using Gumbel Tree-LSTM-based sentence encoders.

Table 1 and 2 contain results on the SNLI task. We can see that models that adopt the proposed ElBiS matching function extract powerful features leading to a performance gain, while keeping similar or less number of parameters. Also, though not directly related to our main contribution, we found that, with elaborated initialization and regularization, simple LSTM models (even the one with the heuristic matching function) achieve competitive performance with those of state-of-the-art models.[4]

4.2 Paraphrase Identification

Another popular task on identifying relationship between a sentence pair is paraphrase identification (PI). The objective of the PI task is to predict whether a given sentence pair has the same meaning or not. To correctly identify the paraphrase relationship, an input to a classifier should contain the semantic similarity and difference between sentences.

For evaluation of paraphrase identification, we

[4] https://nlp.stanford.edu/projects/snli

Matching Fn.	# Params.	Acc. (%)
Concat	1.34M	85.0
Heuristic	1.34M	87.0
ElBiS ($M = 1$)	1.04M	86.7
ElBiS ($M = 2$)	1.35M	**87.3**
ElBiS ($M = 3$)	1.66M	87.1

Table 3: Results on the PI task using LSTM-based sentence encoders.

use Quora Question Pairs dataset[5]. The dataset contains 400k question pairs, each of which is annotated with a label indicating whether the questions of the pair have the same meaning. To our knowledge, the Quora dataset is the largest available dataset of paraphrase identification. We used the same training, development, test splits as the ones used in Wang et al. (2017).

For experiments with heuristic matching, we used the function proposed by Ji and Eisenstein (2013), which is shown by the authors to be effective in matching vectors in latent space compared to simple concatenation. It is composed of the element-wise product and absolute difference between two vectors: $[\mathbf{a} \odot \mathbf{b}; |\mathbf{a} - \mathbf{b}|]$, where $\mathbf{a}$ and $\mathbf{b}$ are encoded sentence vectors.

Similar to NLI experiments, we use a single layer unidirectional LSTM network as sentence encoder, and we state detailed settings in §A.2. The results on the PI task is listed in Table 3. Again we can see that the models armed with the ElBiS matching function discover parsimonious and effective interaction between vectors.

5 Conclusion and Discussion

In this work, we propose ElBiS, a general method of fusing information from two sentence vectors. Our method does not rely on heuristic knowledge constructed for a specific task, and adaptively learns from data the element-wise connections between vectors from data. From experiments, we demonstrated that the proposed method outperforms or matches the performance of commonly used concatenation-based or heuristic-based feature functions, while maintaining the fused representation compact.

Although the main focus of this work is about sentence matching, the notion of element-wise bilinear interaction could be applied beyond sentence matching. For example, many models that specialize in NLI have components where the heuristic matching function is used, e.g. in computing intra-sentence or inter-sentence attention weights. It could be interesting future work to replace these components with our proposed matching function.

One of the main drawback of our proposed method is that, due to its improved expressiveness, it makes a model overfit easily. When evaluated on small datasets such as Sentences Involving Compositional Knowledge dataset (SICK; Marelli et al., 2014) and Microsoft Research Paraphrase Corpus (MSRP; Dolan and Brockett, 2005), we observed performance degradation, partly due to overfitting. Similarly, we observed that increasing the number of interaction types M does not guarantee consistent performance gain. We conjecture that these could be alleviated by applying regularization techniques that control the sparsity of interaction, but we leave it as future work.

Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea Government (MSIT) (NRF-2016M3C4A7952587).

References

Aishwarya Agrawal, Jiasen Lu, Stanislaw Antol, Margaret Mitchell, C. Lawrence Zitnick, Devi Parikh, and Dhruv Batra. 2017. Vqa: Visual question answering. *International Journal of Computer Vision*, 123(1):4–31.

Hedi Ben-younes, Remi Cadene, Matthieu Cord, and Nicolas Thome. 2017. MUTAN: Multimodal tucker fusion for visual question answering. In *The IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. 2015. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 632–642, Lisbon, Portugal. Association for Computational Linguistics.

Samuel R. Bowman, Jon Gauthier, Abhinav Rastogi, Raghav Gupta, Christopher D. Manning, and Christopher Potts. 2016. A fast unified model for parsing and sentence understanding. In *Proceedings of the 54th Annual Meeting of the Association*

[5] https://data.quora.com/First-Quora-Dataset-Release-Question-Pairs

for Computational Linguistics (Volume 1: Long Papers), pages 1466–1477, Berlin, Germany. Association for Computational Linguistics.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. 2002. Finding frequent items in data streams. In *International Colloquium on Automata, Languages, and Programming*, pages 693–703, Málaga, Spain. Springer.

Qian Chen, Xiaodan Zhu, Zhen-Hua Ling, Si Wei, Hui Jiang, and Diana Inkpen. 2017. Recurrent neural network-based sentence encoder with gated attention for natural language inference. In *Proceedings of the 2nd Workshop on Evaluating Vector Space Representations for NLP*, pages 36–40. Association for Computational Linguistics.

Jihun Choi, Kang Min Yoo, and Sang-goo Lee. 2018. Learning to compose task-specific tree structures. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, New Orleans, Louisiana, USA. Association for the Advancement of Artificial Intelligence.

George Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals and Systems*, 2(4):303–314.

William B. Dolan and Chris Brockett. 2005. Automatically constructing a corpus of sentential paraphrases. In *Proceedings of the Third International Workshop on Paraphrasing (IWP 2005)*, pages 9–16, Jeju Island, Korea. Asian Federation of Natural Language Processing.

Akira Fukui, Dong Huk Park, Daylen Yang, Anna Rohrbach, Trevor Darrell, and Marcus Rohrbach. 2016. Multimodal compact bilinear pooling for visual question answering and visual grounding. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 457–468, Austin, Texas. Association for Computational Linguistics.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1026–1034, Washington, DC, USA. IEEE Computer Society.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Kurt Hornik. 1991. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France. PMLR.

Yangfeng Ji and Jacob Eisenstein. 2013. Discriminative improvements to distributional sentence similarity. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 891–896, Seattle, Washington, USA. Association for Computational Linguistics.

Jin-Hwa Kim, Kyoung-Woon On, Woosang Lim, Jeonghee Kim, Jung-Woo Ha, and Byoung-Tak Zhang. 2017. Hadamard product for low-rank bilinear pooling. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations*, San Diego, California, USA.

Ben Krause, Liang Lu, Iain Murray, and Steve Renals. 2016. Multiplicative LSTM for sequence modelling. *arXiv preprint arXiv:1609.07959*.

Tsung-Yu Lin, Aruni RoyChowdhury, and Subhransu Maji. 2015. Bilinear CNN models for fine-grained visual recognition. In *The IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile.

Marco Marelli, Luisa Bentivogli, Marco Baroni, Raffaella Bernardi, Stefano Menini, and Roberto Zamparelli. 2014. SemEval-2014 task 1: Evaluation of compositional distributional semantic models on full sentences through semantic relatedness and textual entailment. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, pages 1–8, Dublin, Ireland. Association for Computational Linguistics and Dublin City University.

Lili Mou, Rui Men, Ge Li, Yan Xu, Lu Zhang, Rui Yan, and Zhi Jin. 2016. Natural language inference by tree-based convolution and heuristic matching. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 130–136, Berlin, Germany. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Andrew M. Saxe, James L. McClelland, and Surya Ganguli. 2014. Exact solutions to the nonlinear dynamics of learning in deep linear neural networks. In *Proceedings of the 2nd International Conference on Learning Representations*, Banff, Alberta, Canada.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Kai Sheng Tai, Richard Socher, and Christopher D. Manning. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1556–1566, Beijing, China. Association for Computational Linguistics.

Zhiguo Wang, Wael Hamza, and Radu Florian. 2017. Bilateral multi-perspective matching for natural language sentences. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence*, pages 4144–4150, Melbourne, Victoria, Australia. International Joint Conferences on Artificial Intelligence.

Wei Wu, Zhengdong Lu, and Hang Li. 2013. Learning bilinear model for matching queries and documents. *Journal of Machine Learning Research*, 14:2519–2548.

Yuhuai Wu, Saizheng Zhang, Ying Zhang, Yoshua Bengio, and Ruslan R Salakhutdinov. 2016. On multiplicative integration with recurrent neural networks. In D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, editors, *Advances in Neural Information Processing Systems 29*, pages 2856–2864. Curran Associates, Inc.

Caiming Xiong, Stephen Merity, and Richard Socher. 2016. Dynamic memory networks for visual and textual question answering. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2397–2406, New York, New York, USA. PMLR.

Dani Yogatama, Phil Blunsom, Chris Dyer, Edward Grefenstette, and Wang Ling. 2017. Learning to compose words into sentences with reinforcement learning. In *Proceedings of the 5th International Conference on Learning Representations*, Toulon, France.

Zhou Yu, Jun Yu, Jianping Fan, and Dacheng Tao. 2017. Multi-modal factorized bilinear pooling with co-attention learning for visual question answering. In *The IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy.

A Experimental Settings

A.1 Natural Language Inference

For all experiments, we used the Adam (Kingma and Ba, 2015) optimizer with a learning rate 0.001 and halved the learning rate when there is no improvement in accuracy for one epoch. Each model is trained for 10 epochs, and the checkpoint with the highest validation accuracy is chosen as final model. Sentences longer than 25 words are trimmed to have the maximum length of 25 words, and batch size of 64 is used for training.

For all experiments, we set the dimensionality of sentence vectors to 300. 300-dimensional GloVe (Pennington et al., 2014) vectors trained on 840 billion tokens[6] were used as word embeddings and not updated during training. The number of hidden units of the single-hidden layer MLP is set to 1024.

Dropout (Srivastava et al., 2014) is applied to word embeddings and the input and the output of the MLP. The dropout probability is selected from $\{0.10, 0.15, 0.20\}$. Batch normalization (Ioffe and Szegedy, 2015) is applied to the input and the output of the MLP.

Recurrent weight matrices are orthogonally initialized (Saxe et al., 2014), and the final linear projection matrix is initialized by sampling from $\text{Uniform}(-0.005, 0.005)$. All other weights are initialized following the scheme of He et al. (2015).

A.2 Paraphrase Identification

For PI experiments, we used the same architecture and training procedures as NLI experiments, except the final projection matrix and heuristic matching function. Also, we found that the PI task is more sensitive to hyperparameters than NLI, so we apply different dropout probabilities to the encoder network and to the classifier network. Both values are selected from $\{0.10, 0.15, 0.20\}$. Each model is trained for 15 epochs, and the checkpoint with the highest validation accuracy is chosen as final model.

[6] `http://nlp.stanford.edu/data/glove.840B.300d.zip`