Siamese vs Non-Siamese: Dual Encoders for Intent Detection

Anonymous ACL submission

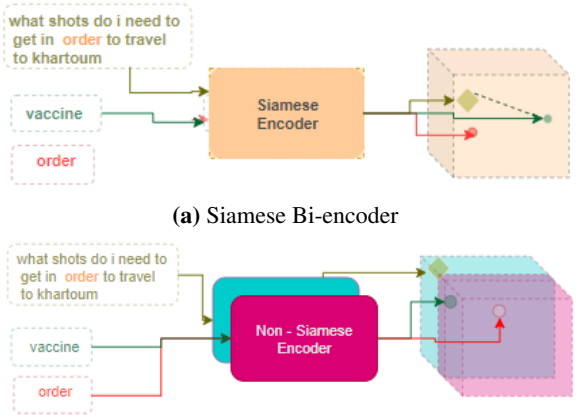
Abstract

Bi-encoders have been shown to be effective for intent classification. Current Bi-encoders use the same weights to learn the embedding of both the contexts and candidates. However, this can be counter-productive when there exist contexts with overlapping keywords from competing candidate labels. This could lead to unrelated context and candidate having similar embeddings and being mis-classified. In this work, we investigate the potential of non-siamese Bi-encoders for intent detection, where separate weights are learned for context and candidate. Our results show that non-siamese Bi-encoders improve the performance of traditional Bi-encoders across datasets. We also show that using heterogeneous architectures in a non-siamese Bi-encoder can effectively reduce memory and computation requirement while maintaining prediction performance.

1 Introduction

Intent classification is the task of classifying a sequence of text (*context*) into one of a set of predefined intents (*candidates*). A promising approach for this task involves the use of Bi-encoders (Reimers and Gurevych, 2019; Casanueva et al., 2020; Clarke et al., 2022) which encode the context and candidate into a latent embedding space and compute the semantic similarity to predict an appropriate candidate.

This dual encoder architecture has the advantage of being more computationally efficient and flexible compared to its cross-encoding and conventional sequence classifier counterparts. At inference time, the candidate embeddings only need to be generated once. Therefore these representations can be cached and subsequently used when inferring future context-candidate pairs, thus reduce the amount of computation required and vastly improve evaluation and inference time (Humeau et al., 2020; Thakur et al., 2020; Geigle et al., 2021). Additionally, since predictions are done via distance comparisons of vector representations, bi-encoders can be easily adapted to novel candidates unseen during training.



(a) Siamese Bi-encoder

(b) Non-Siamese Bi-encoder

Figure 1: Siamese vs non-siamese Bi-encoder for intent classification

Typically, dual encoders utilize a siamese network structure where the models for the context and candidate encoders have tied weights, as shown in Figure 1a. This is desirable when the two inputs share many common traits, e.g., image matching and voice recognition (Zhang and Duan, 2017; Han et al., 2015). However, when applied to the nuanced task of intent detection, this siamese structure can be counter-productive for generating truly representative and accurate vector representations, as contexts can contain meaningful and impactful tokens from conflicting candidates and vice versa. Figure 1 shows a real example from the Clinc-150 dataset: the context sentence contains the key (and only) word in a competing candidate label (*order*). As such, tying the network weights can potentially result in context representations being overly and incorrectly fit to conflicting candidates (Figure 1a), causing mis-classification.

In this work, we investigate the potential of learning separate weights for context and candidate in Bi-encoders for intent classification. We study the non-siamese Bi-encoder architecture, where the weights are trained and updated through two sepa-

rate back-propagation passes for context and candidate during training. Our key intuition is that by learning the two embedding spaces independently, we can learn more accurate semantic relationships between contexts and candidates pairs (Figure 1b).

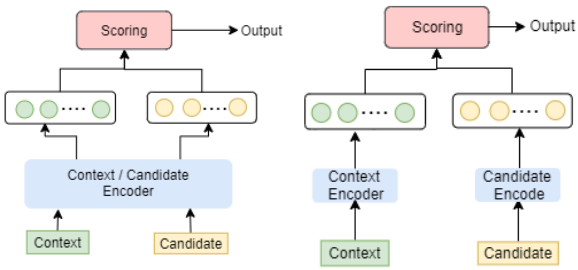
One consideration of non-siamese Bi-encoders is the additional memory requirement for two encoder models. To alleviate this, we explore heterogeneous encoder models for context and candidate, where one of the encoders is a lightweight transformer (e.g. TinyBert).

To study the performance of non-siamese and heterogeneous Bi-encoders, we use 4 representative intent classification datasets. Additionally, to further study the phenomenon of conflicting context and candidates, we create new *borderline* versions of the 4 datasets representing the more challenging case of conflicting context candidate pairs. Our experimental results show that non-siamese Bi-encoders outperforms siamese Bi-encoders by up to 4.6% on the vanilla datasets and 8.7% on the *borderline* versions. We also show that heterogenous Bi-encoder perform similarly or better than homogeneous non-siamese Bi-encoders while reducing memory and computation requirement.

2 Related Work

The task of classifying a given input context to a candidate label is a well-studied in machine learning. A promising approach to this problem is Bi-Encoder. The core idea of Bi-encoders is to map the input and a candidate label separately into a common dense vector space and perform scoring via a distance metric such as dot product or cosine similarity (Wu et al., 2017; Reimers and Gurevych, 2019; Mithun et al., 2018; Jung et al., 2022). A major advantage of Bi-encoder architecture is its computation efficiency during evaluation, because of its ability to cache the representations of the candidates.

Typical Bi-encoder models use the siamese structure (Bromley et al., 1993) where the models contain two or more identical sub-networks. This approach has shown impressive results across a range of tasks such as face recognition, speech processing and informational retrieval (Han et al., 2015; Zhang and Duan, 2017; Jung et al., 2022; Clarke et al., 2022). However, when applied to a more nuanced task such as intent classification where context and candidate differ in length and have the potential for large amounts of meaningful token overlap, it makes sense to allow a degree of deviation between the context and candidate. As such



(a) Siamese **(b)** Non-Siamese

Figure 2: Arch. of siamese and non-siamese Bi-encoder.

we explore the use of non-siamese networks and heterogeneous bi-encoders to address these issues.

3 Method

In this section, we formulate using Bi-encoders for intent classification and describe the siamese, non-siamese and heterogenous architectures.

3.1 Problem Formulation

The intent classifications task defines a set of candidate categories $\mathcal{Y} = \{y_i\}_n^1$. The task is to map a given sequence of text x (context) into one of the candidate categories. During inference using Bi-encoder, each text and label pair $\{(x, y_i) \mid y_i \in \mathcal{Y}\}$ is encoded by their respective models. We define the context encoder as Φ_t and the candidate encoder as Φ_l, which produces vectors $\vec{t}, \vec{l} \in \mathbb{R}^d$ for input context and candidate, respectively. Similarity score is computed for each (x, y_i) pair with a distance metric θ and the candidate label y_i closest to the context text x in the embedding space is selected as the prediction, as follows:

$$\hat{y} = \underset{y_i \in \mathcal{Y}}{\arg\max}\, \theta \left(\Phi_t^{\mathcal{W}_t}(x), \Phi_l^{\mathcal{W}_l}(y_i) \right) \quad (1)$$

where $\mathcal{W}_t, \mathcal{W}_l$ are the weights for encoders Φ_t and Φ_l, respectively.

3.2 Siamese Bi-encoders

Siamese networks such as the one depicted in Figure 2a are trained with tied weights. The same network Φ embeds the context and candidates, so $\Phi \equiv \Phi_t \equiv \Phi_l$ and $\mathcal{W} \equiv \mathcal{W}_t \equiv \mathcal{W}_l$. During training, the gradients of the candidate and context accumulate and then update the same network:

$$\partial\Phi = \text{BackProp}\,(x) + \text{BackProp}\,(y_i) \quad (2)$$

3.3 Non-Siamese Bi-encoders

As shown in Figure 2b, non-siamese Bi-encoders use separate dedicated networks for context and

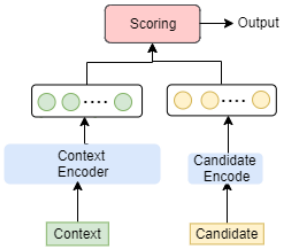


Figure 3: Heterogeneous Bi-encoder

candidate, i.e., Φ_t and Φ_l are separate instances with independent weights. In contrast to siamese networks, non-siamese networks learn the representation for context and candidate independently. We define the *homogeneous* non-siamese Bi-encoders, where the two encoders share the same architecture but with distinct weights, as in, $\Phi_t = \Phi_l$ and $\mathcal{W}_t \neq \mathcal{W}_l$. The inference process is the same as defined in Equation 1. During training, the network weights are updated independently:

$$\begin{aligned} \partial\Phi_t &= \text{BackProp}\,(x) \\ \partial\Phi_l &= \text{BackProp}\,(y_i) \end{aligned} \tag{3}$$

3.4 Heterogeneous Bi-encoder

One drawback for non-siamese Bi-encoder is the additional memory required for the second encoder model. To alleviate this, we explore *heterogeneous* Bi-encoders. Figure 3 shows this design where the two encoders can vary in model architecture and size. This opens up a significant design space for mixing and matching different encoder models and the opportunity to use a more lightweight model for one of the encoders to reduce memory consumption. For heterogeneous Bi-encoders, $\Phi_t \neq \Phi_l$ and $\mathcal{W}_t \neq \mathcal{W}_l$.

3.5 Training Bi-encoders

We investigate the optimal training strategy using Bi-encoders for intent classification. Specifically, we investigate the implication of 1) loss function design, 2) distance metric and 3) data augmentation. For loss function, we consider contrastive loss and mean-squared-error (MSE) loss. We compare the impact of the distant metrics consine similarity and dot product. Lastly, we experiment with training using only positive examples and augmenting with negative examples.

4 Experimental Setup

Dataset We select 4 intent classification datasets, with a wide range of label counts (7 - 150) and diverse domains including banking, travel and dining (Table 1).

Dataset	Description	#Label	Train	Test
Banking77	Online banking queries	77	10K	3.1K
Clinc-150	Virtual assistants in production	150	15K	4.5K
SGD	Task-oriented conversations	34	16K	4.2K
SNIPS	Smart assistants questions	7	11K	4.0K

Table 1: Datasets

Datasets	% no-overlap w/ ground truth	% overlap w/ negative candidates	Total Size
Banking77	8.6	8.3	10K
Clinc-150	14.8	12.5	15K
SGD	4.6	4.6	16K
SNIPS	26.4	8.4	11K

Table 2: Borderline versions of the datasets. Percentages of examples sub-sampled at each stage are shown here, as well as the total number of sub-sampled examples.

Borderline Dataset Construction To investigate the impact of the word overlap between conflicting context and candidate on Bi-encoder performance, we construct a new version of the 4 datasets through a 2-stage sub-sampling process: **1)** we remove the context utterances that share tokens with its ground-truth candidate label; and **2)** we select the context utterances with tokens from competing (non ground-truth) candidate labels, which we refer to as Borderline examples. Table 2 summarizes this process.

Implementation We implement our Bi-encoder based on the design and open-source code of SBERT (Reimers and Gurevych, 2019). We use BERT (Devlin et al., 2018) as the encoder model unless specified otherwise. We use dimension-wise mean as the pooling operation. For training, we use the Adam optimizer (Kingma and Ba, 2015) with weight decay of 0.01, batch size of 16 and 50 epochs. We use linear learning rate warm-up over the first 10% steps and a linear schedule.

5 Results and Discussion

5.1 Training a Bi-encoder

Table 3 shows the accuracy of a siamese Bi-encoder trained using different loss function and similarity metrics and trained on dataset with positive-only examples vs. positive+negative examples. We draw several key insights. First, we observe that the models do not learn effectively when using cosine scoring with positive-only examples, especially on the Clinc-150 dataset. When the training data is augmented with negative examples, performance improves significantly across datasets. Second, the design of loss function and distance metric need to be considered jointly. Specifically, MSE performs better with cosine scoring while contrastive loss

Dataset	Loss Function	Similarity Metric	+ive Only	+ive & -ive
SNIPS	MSE	cos	82.9	**97.9**
		dot	24.7	**97.7**
	Contrastive	cos	78.9	**98.4**
		dot	**98.4**	98.1
Clinc-150	MSE	cos	33.6	**76.8**
		dot	28.4	**46.7**
	Contrastive	cos	34.5	**77.7**
		dot	**89.4**	89.1

Table 3: Configuration search results on siamese TinyBERT (Jiao et al., 2020), trained with +ive (positive) only and (+ive plus -ive) negative sampling.

Model Type	Banking77	Clinc-150	SGD	SNIPS
Siamese	87.3	86.5	74.3	**97.8**
Non-Siamese	**88.8**	**88.9**	**77.7**	97.6

Table 4: Siamese vs Non-Siamese Accuracy Score with contrastive loss and dot distance metric trained on only positive samples

Model Type	Banking77	Clinc-150	SGD	SNIPS
Siamese	69.3	72.9	**83.8**	**97.2**
Non-Siamese	**75.3**	**76.5**	83.1	95.1

Table 5: Siamese vs Non-Siamese Accuracy Score with contrastive loss and dot distance metric trained on BLCD dataset

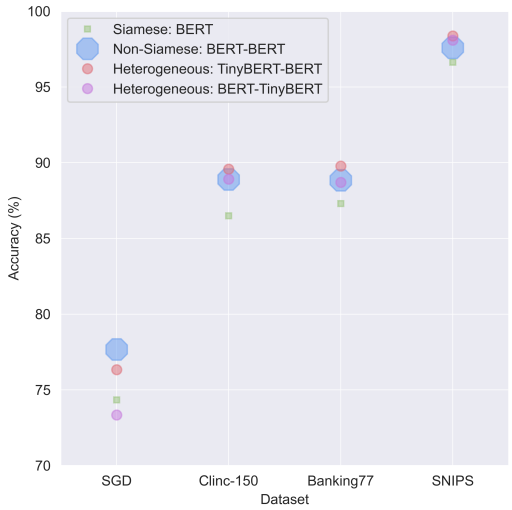


Figure 4: Heterogeneous Bi-encoder results with contrastive loss and dot product. For Non-Siamese setups, separated by the dash, the legend shows the base encoder for context and candidate respectively. The marker sizes correspond to model # parameters.

performs best with dot product. Third, when comparing across all three design choices, we find that bi-encoders perform the best with contrastive loss function and dot product distance metric training on positive-only examples. This conclusion holds true for non-siamese Bi-encoder as well (Table 6).

5.2 Non-Siamese vs Siamese

Table 4 shows the accuracy of non-siamese Bi-encoder with traditional siamese Bi-encoder. We observe that learning representations for the context and candidate separately improves the performance of Bi-encoder for 3 of the 4 datasets. Table 5 shows the comparison of non-siamese and siamese on the more challenging Borderline datasets described in Section 4. Borderline examples are contexts that contain meaningful and impactful tokens from competing, non-ground-truth candidate labels. We observe a larger accuracy gap between non-siamese and siamese on the borderlines examples of Banking77 and Clinc-150, while they perform comparably on SGD and SNIPS. Banking77 and Clinc-150 have significantly more borderline candidate labels (32 and 80) than SGD and SNIPS (11 and 6) and our intuition is that a larger and more diverse candidate pool creates a more challenging context/candidate conflicting scenario.

5.3 Exploring the Heterogeneous Bi-encoder

We experiment with heterogeneous Bi-encoders by configuring one of the encoders as Bert-base and the other as TinyBERT. BERT-base and TinyBERT differ in model size (12 vs 2 layers) and output embedding size (768 vs. 128). We reduce the hidden dimension size of BERT-base encoder to match that of the TinyBERT encoder. We compare the classification accuracy and model size of siamese, homogeneous and heterogeneous non-siamese Bi-encoders in Figure 4, where the marker size represents the size of the overall model. We observe that the two heterogeneous configurations perform similarly or better than the homogeneous non-siamese models while requiring significantly less memory (91.5%). In addition, heterogeneous Bi-encoders allow different sequences length for context and candidate. This shows heterogeneous Bi-encoders is a promising architecture design to further improve the performance and practicality of Bi-encoders and it can benefit from further studies.

6 Conclusion

In this paper, we study the potential of Non-Siamese Bi-encoders for intent classification. We highlight the power of independently learning sentence representations and it ability to resolve challenges cases of meaningful token overlap in content candidate pairs. We also show that heterogenous Bi-encoder perform similarly or better than homogeneous non-siamese Bi-encoders while reducing memory and computation requirement.

References

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a "siamese" time delay neural network. In *Advances in Neural Information Processing Systems*, volume 6. Morgan-Kaufmann.

Iñigo Casanueva, Tadas Temcinas, Daniela Gerz, Matthew Henderson, and Ivan Vulic. 2020. Efficient intent detection with dual sentence encoders. In *Proceedings of the 2nd Workshop on NLP for ConvAI - ACL 2020*. Data available at https://github.com/PolyAI-LDN/task-specific-datasets.

Christopher Clarke, Joseph Peper, Karthik Krishnamurthy, Walter Talamonti, Kevin Leach, Walter Lasecki, Yiping Kang, Lingjia Tang, and Jason Mars. 2022. One agent to rule them all: Towards multi-agent conversational AI. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 3258–3267, Dublin, Ireland. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

Gregor Geigle, Jonas Pfeiffer, Nils Reimers, Ivan Vulić, and Iryna Gurevych. 2021. Retrieve fast, rerank smart: Cooperative and joint approaches for improved cross-modal retrieval.

Xufeng Han, Thomas Leung, Yangqing Jia, Rahul Sukthankar, and Alexander C. Berg. 2015. Matchnet: Unifying feature and metric learning for patch-based matching. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3279–3286.

Samuel Humeau, Kurt Shuster, Marie-Anne Lachaux, and Jason Weston. 2020. Poly-encoders: Architectures and pre-training strategies for fast and accurate multi-sentence scoring. In *International Conference on Learning Representations*.

Xiaoqi Jiao, Yichun Yin, Lifeng Shang, Xin Jiang, Xiao Chen, Linlin Li, Fang Wang, and Qun Liu. 2020. TinyBERT: Distilling BERT for natural language understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 4163–4174, Online. Association for Computational Linguistics.

Euna Jung, Jaekeol Choi, and Wonjong Rhee. 2022. Semi-siamese bi-encoder neural ranking model using lightweight fine-tuning. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 502–511, New York, NY, USA. Association for Computing Machinery.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Niluthpol Chowdhury Mithun, Juncheng Li, Florian Metze, and Amit K. Roy-Chowdhury. 2018. Learning joint embedding with multimodal cues for cross-modal video-text retrieval. In *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval*, ICMR '18, page 19–27, New York, NY, USA. Association for Computing Machinery.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nandan Thakur, Nils Reimers, Johannes Daxenberger, and Iryna Gurevych. 2020. Augmented sbert: Data augmentation method for improving bi-encoders for pairwise sentence scoring tasks.

Ledell Wu, Adam Fisch, Sumit Chopra, Keith Adams, Antoine Bordes, and Jason Weston. 2017. Starspace: Embed all the things!

Yichi Zhang and Zhiyao Duan. 2017. Iminet: Convolutional semi-siamese networks for sound search by vocal imitation. In *2017 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 304–308.

A Training configuration for siamese and non-siamese Bi-encoders

Table 6 shows the complete results for investigating the best training strategy for Bi-encoders. We experiment with loss function, distance metrics, positive-only vs. positive + negative training examples and varying learning rates. This table compliments the results shown in Table 3 and discussion in 5.1.

Hyperparameter			Siamese		Non-Siamese	
Learning Rate	Loss Function	Similarity Metric	SNIPS	Clinc-150	SNIPS	Clinc-150
1e-2	MSE	cos	54.6 / 13.2	30.0 / 0.8	60.0 / **20.1**	**5.7** / 0.7
		dot	28.5 / 96.2	5.9 / 13.6	14.2 / 14.1	0.2 / 0.7
	Contrastive	cos	52.9 / 15.0	23.8 / **0.9**	68.8 / 8.7	4.9 / 0.7
		dot	**71.5** / **15.1**	**44.0** / 0.7	**65.9** / 15.1	0.7 / **2.0**
1e-3	MSE	cos	71.7 / **98.1**	29.5 / 68.5	86.2 / 98.1	3.4 / 66.4
		dot	45.3 / 97.7	32.2 / 58.1	34.8 / 97.7	1.3 / 32.4
	Contrastive	cos	75.4 / **98.1**	30.6 / 80.1	84.9 / 98.1	2.6 / 81.9
		dot	**98.6** / 97.5	**89.3** / **89.2**	**98.1** / **98.4**	**90.0** / **90.3**
1e-4	MSE	cos	82.9 / 97.9	33.6 / 76.8	86.8 / 98.1	5.2 / 77.9
		dot	24.7 / 97.7	28.4 / 46.7	19.3 / 97.8	6.8 / 22.9
	Contrastive	cos	78.9 / **98.4**	34.9 / 77.7	85.3 / **98.6**	3.6 / 78.8
		dot	**98.4** / 98.1	**89.1** / **89.4**	**98.4** / 98.4	**90.0** / **90.5**

Table 6: Configuration search results on TinyBERT (Jiao et al., 2020). Separated by a slash, each column contains test ac curacies when trained on positive pairs only and trained with negative sampling.