Chapter 4

Siamese Neural Networks for STS

A *Siamese Neural Network* refers to a class of neural network architecture that contains two or more identical subnetworks. This is usually employed in applications that require comparisons between two or more inputs (signature verification, image similarity etc.). The network can take two or more inputs at the same time and they will be processed by different subnetworks simultaneously. The subnetworks have the same configuration with identical parameters and weights. The training process is mirrored across all of the subnetworks, meaning that the parameters remain constant across all of the subnetworks. Each subnetwork contains a traditional perceptron model such as LSTM, CNN etc. The neural network compares the outputs of the two subnetworks through a distance metric such as cosine distance, and the error is propagated back (Mueller and Thyagarajan 2016). In the testing phase, the neural network predicts whether the two inputs are different through this similarity measure (Neculoiu et al. 2016).

Siamese neural networks have been employed in many applications in different areas such as signal processing (Thiolliere et al. 2015; Manocha et al. 2018; Shon et al. 2017; Zhang et al. 2019b; Švec et al. 2017; Gündoğdu et al. 2017;

Siddhant et al. 2017; Zeghidour et al. 2016), biology (Zheng et al. 2018; Szubert et al. 2019), chemistry and pharmacology (Jeon et al. 2019), geometry (Sun et al. 2020b), computer vision (Baldi and Chauvin 1993; Chopra et al. 2005; He et al. 2018; Paisios et al. 2012; Yi et al. 2014; Lefebvre and Garcia 2013; Taigman et al. 2014; Berlemont et al. 2015; Kassis et al. 2017; Hanif 2019), physics (Zou et al. 2018; De Baets et al. 2019), robotics (Utkin et al. 2017a; Utkin et al. 2017b; Zeng et al. 2018), video processing (Ryoo et al. 2018; Liu et al. 2018b; Liu et al. 2018a; Lee and Kim 2019) etc. They have also been used in NLP tasks (Yih et al. 2011; Kumar et al. 2018; González et al. 2019) including STS (Das et al. 2016; Neculoiu et al. 2016; Mueller and Thyagarajan 2016). In fact, the best result from the SICK dataset (Marelli et al. 2014) was provided by a Siamese neural network (Mueller and Thyagarajan 2016) which shows that state-of-the-art in STS are Siamese neural networks.

In addition to providing state-of-the-art results in STS, there are additional advantages of using Siamese neural networks. As previously mentioned, in Siamese neural networks the weights are shared across subnetworks resulting in fewer parameters to train, this in turn means that less training data is required and there is a lower likelihood of overfitting (Ranasinghe et al. 2019b). Given the amount of human labour required to produce datasets for STS, Siamese neural networks can provide an ideal solution. Another advantage is that when trained on a STS task the Siamese neural network architecture can be adapted as a sentence encoder. The output vector of the subnetwork is a semantically rich vector representation of the input sentence (Mueller and Thyagarajan 2016).

These advantages have motivated us to further explore Siamese neural network architectures in STS.

We address four research questions in this chapter:

RQ1: Can an existing state-of-the-art Siamese neural network architecture be modified to provide better STS results?

RQ2: Can the method be further improved with transfer learning and data augmentation techniques?

RQ3: Can the proposed Siamese neural network be easily adapted for different languages?

RQ4: How well does the proposed Siamese neural network perform in a different domain?

The main contributions of this chapter are as follows.

1. We propose a GRU (Gated Recurrent Unit) based Siamese neural network that outperformed the state-of-the-art LSTM based Siamese neural network in small STS datasets.

2. We propose an LSTM (Long Short Term Memory) and self-attention based Siamese neural network that outperformed the state-of-the-art LSTM based Siamese neural network in large STS datasets.

3. We propose further enhancements to the architecture using transfer learning and data augmentation.

4. We evaluate how well the proposed Siamese neural network architecture performs in different languages and domains.

5. The code and the pre-trained models are publicly available to the community[1].

The rest of this chapter is organised as follows. Section 4.1 describes the existing research done on Siamese neural networks. Section 4.2 discusses the methodology and the experiments undertaken with three English STS datasets. Sections 4.2.1 and 4.2.2 provide more experiments which improve the results. Experiments conducted with other languages and domains are shown in Sections 4.3 and 4.4. The chapter finishes with conclusions and ideas for future research directions in Siamese neural networks.

4.1 Related Work

The Siamese neural networks are prevalent in the machine learning community. The first appearance of Siamese neural networks date back to 1993 when they were first introduced by Bromley et al. (1993) to detect forged signatures. By comparing two handwritten signatures, this Siamese neural network could predict if the two signatures were both original or if one was a forgery. Preceding this, Baldi and Chauvin (1993) introduced a similar artificial neural network able to recognise fingerprints, though by a different name.

Siamese neural networks have been employed for various uses since these initial applications. In the audio and speech signal processing field, Thiolliere et al. (2015) merged a dynamic-time warping based spoken term discovery (STD)

[1] The public GitHub repository is available on https://github.com/tharindudr/Siamese-recurrent-rrchitectures

system with a Siamese deep neural network for the automatic discovery of linguistic units from raw speech. Manocha et al. (2018) used a Siamese model to detect all semantically similar audio clips from an input audio recording, and Shon et al. (2017) employed a Siamese model to recognise Arabic dialects from the Arabic speech content found in media broadcasts. In biology, Zheng et al. (2018) implemented a Siamese neural network to compare DNA sequences and recently Szubert et al. (2019) presented a Siamese neural network-based technique for a visualisation and interpretation of single-cell datasets. Image analysis is the field with the highest number of applications for the Siamese neural networks. Recognising fingerprints (Baldi and Chauvin 1993), similar image detection (Chopra et al. 2005; He et al. 2018; Paisios et al. 2012; Yi et al. 2014; Lefebvre and Garcia 2013), face verification (Taigman et al. 2014), gesture recognition (Berlemont et al. 2015), handwriting analysis (Kassis et al. 2017) and patch matching (Hanif 2019) are some examples of them. All of these tasks involve the comparison of two or more things.

Recently, Siamese neural networks have also been employed in NLP. Yih et al. (2011) proposed similarity Learning via Siamese Neural Network (S2Net), a technique able to discriminatingly learn the concept vector representations of text words. Kumar et al. (2018) used a Siamese neural network to recognise clickbaits in online media outlets. González et al. (2019) proposed a natural language processing application of the Siamese neural network for extractive summarisation, which means that their technique can extrapolate the most relevant sentences in a document.

Siamese neural networks are not limited to these applications and have also been implemented in STS tasks in NLP. Das et al. (2016) used a CNN based Siamese neural network to detect similar questions on question and answer websites such as Yahoo Answers, Baidu, Zhidao, Quora, and Stack Overflow. Neculoiu et al. (2016) employed a Siamese neural network based on Bidirectional LSTMs to identify similar job titles. The baseline we used for this chapter, MALSTM (Mueller and Thyagarajan 2016) uses LSTM based Siamese neural network to produce semantic textual similarity, and it provides the best results for the SICK dataset outperforming other STS methods such as Tree-LSTMs (Tai et al. 2015). They use the exponent of the negative Manhattan distance between two outputs from the two subnetworks as the similarity function. Due to the performance, this can be considered as the state-of-the-art Siamese neural network for STS. However, this architecture leaves considerable room for variation, which we exploit in this chapter as we explain in Section 4.2.

4.2 Exploring Siamese Neural Networks for STS

The basic structure of the Siamese neural network architecture used in our experiments is shown in Figure 4.1. It consists of an embedding layer that represents each sentence as a sequence of word vectors. This sequence of word vectors is then fed into a Recurrent Neural Network (RNN) cell, which learns a mapping from the space of variable-length sequences of 300-dimensional vectors into a 50 dimensional vector. The sole error signal backpropagated during training stems from the similarity between these 50 dimensional vectors, which can also be used as a sentence representation. Initially, the similarity function we used was based on Manhattan distance. To ensure that the prediction is between 0 and 1, we took the exponent of the negative Manhattan distance between two sentence representations. The similarity function was adapted from Mueller and Thyagarajan (2016). The proposed variants of our architecture are:

1. LSTM - Block A in Figure 4.1 contains a single LSTM cell. This is the architecture suggested by Mueller and Thyagarajan (2016)

2. Bi-directional LSTM - Block A in Figure 4.1 contains a single Bi-directional LSTM cell. Bi-directional LSTM tends to understand the context better than Unidirectional LSTM (Schuster and Paliwal 1997).

3. GRU - Block A in Figure 4.1 contains a single GRU cell. GRUs have been shown to exhibit a better performance on smaller datasets (Chung et al. 2014).

4. Bi-directional GRU - Block A in Figure 4.1 contains a single Bi-directional GRU cell. Bi-directional GRUs tend to understand the context better than Unidirectional GRUs (Vukotić et al. 2016).

5. LSTM + Attention - Block A in Figure 4.1 contains a single LSTM cell with self attention (Vaswani et al. 2017a).

6. GRU + Attention - Block A in Figure 4.1 contains a single GRU cell with self attention (Vaswani et al. 2017a).

7. GRU + Capsule + Flatten - Block A in Figure 4.1 contains a GRU followed by a capsule layer and a flatten layer. Dynamic routing used between capsules performs better than a traditional max-pooling layer (Sabour et al. 2017).

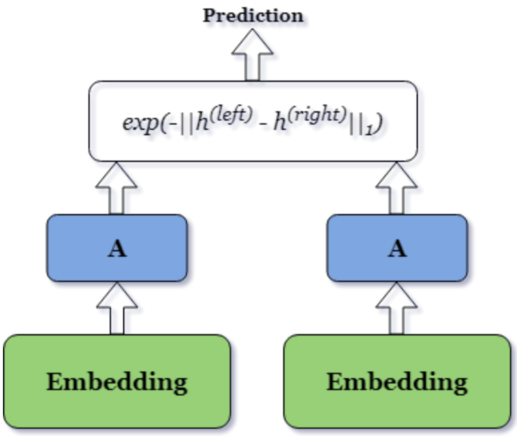

Figure 4.1: Basic structure of the Siamese neural network. Unit A is changed over the architectures.

As the word embedding model, we used Word2vec embeddings (Mikolov et al. 2013a) pre-trained on the Google news corpus[2]. Using this model, we represented each word as a 300 length vector. For the words that do not appear in the model, we used a random vector. We evaluated all of the above variations in the three English STS datasets we introduced in 1; SICK, STS 2017 and QUORA. We trained the Siamese models on the training sets in these datasets and evaluated them on the testing sets. The results are shown in Table 4.1, Table 4.2 and Table 4.3 respectively.

As can be seen in Tables 4.1 and 4.2, for the SICK and STS2017 datasets,

[2] Pre-trained Word2vec can be downloaded from https://code.google.com/archive/p/word2vec/

Model	ρ	τ
LSTM	0.802	0.733
Bi-LSTM	0.784	0.708
GRU	$0.838^\dagger$	$0.780^\dagger$
Bi-GRU	0.832	0.773
LSTM + Attention	0.827	0.765
GRU + Attention	0.818	0.751
GRU + Capsule + Flatten	0.806	0.733

Table 4.1: Results for SICK dataset with different variants of Siamese Neural Network. For each variant, Pearson Correlation (ρ) and Spearman Correlation (τ) are reported between the predicted values and the gold labels of the test set. The best result from all the variations is marked with †.

Model	ρ	τ
LSTM	0.831	0.762
Bi-LSTM	0.784	0.708
GRU	$0.853^\dagger$	$0.811^\dagger$
Bi-GRU	0.844	0.804
LSTM + Attention	0.830	0.791
GRU + Attention	0.825	0.782
GRU + Capsule + Flatten	0.806	0.765

Table 4.2: Results for STS 2017 dataset with different variants of Siamese Neural Network. For each variant, Pearson Correlation (ρ) and Spearman Correlation (τ) are reported between the predicted values and the gold labels of the test set. The best result from all the variations is marked with †.

the GRU based Siamese neural network model outperformed the LSTM based Siamese neural network model, which we used as a baseline, and this provided the best result. It can be seen that complex architectures that involve Bi-directional RNNs, Attention and Capsule mechanisms did not perform well when compared to simple architectures like GRU. We can conclude that for the smaller datasets like STS 2017 and SICK, the GRU based architecture performs better because GRU has fewer parameters than LSTM (Chung et al. 2014). With fewer parameters, the architecture does not need many training instances to optimise

Model	RMSE
LSTM	0.412
Bi-LSTM	0.402
GRU	0.415
Bi-GRU	0.408
LSTM + Attention	$0.382^{\dagger}$
GRU + Attention	0.398
GRU + Capsule + Flatten	0.421

Table 4.3: Results for QUORA dataset with different variants of Siamese Neural Network. For each variant, Root Mean Squared Error (RMSE) reported between the predicted values and the gold labels of the test set. The best result from all the variations is marked with †.

the weights during the training process.

However, when it comes to the big STS dataset, QUORA, the way the variants of the Siamese neural network behaves is different. As we introduced in Chapter 1, QUORA was the biggest STS dataset we experimented with, and it has 320,000 training instances. As a result, even complex architectures like RNNs with Attention get the opportunity to optimise their parameters and deliver good results. This can be seen in Table 4.3. For the QUORA dataset, the LSTM + Attention based Siamese neural network model outperformed the LSTM based Siamese neural network model, which we used as a baseline, and provided the best result. For bigger datasets, we can conclude that Siamese neural networks based on LSTM with Attention would outperform Siamese neural networks only with LSTMs.

From the variants we examined, one notable observation is the poor performance of capsules in Siamese architectures. Despite providing good results in many NLP tasks such as text classification (Sabour et al. 2017; Hettiarachchi and Ranasinghe 2019; Xia et al. 2018; Srivastava et al. 2018) and relation extraction (Zhang et al. 2019a; Zhang et al. 2018; Zhang and Geng 2020), the capsule-based variant failed to outperform the simple LSTM based variant even in the bigger STS dataset. This observation implies that capsule-based Siamese neural networks will not be a good fit for STS tasks.

With these findings, we answer our **RQ1** in this chapter. We have improved the state-of-the-art Siamese neural network architecture and propose a GRU based Siamese neural network architecture for the smaller STS datasets and LSTM + Attention based Siamese neural network for larger STS datasets.

4.2.1 Impact of Transfer Learning

Transfer learning is a machine learning method where a model developed for a task is reused as the starting point for a model on a second task. It is a popular approach in deep learning where pre-trained models are used as the starting point for a new task. This is usually done in scenarios where there is not enough data to train a neural network, so that starting from already finetuned weights would be advantageous (Houlsby et al. 2019; Ruder et al. 2019). Transfer learning has often provided good results for smaller datasets. Therefore, we explored the impact of transfer learning with Siamese neural networks in STS.

We saved the weights of the models that were trained on each STS dataset; SICK, STS 2017 and QUORA. We specifically used the two models that performed best in these datasets; Siamese neural network with GRU and the Siamese neural network with LSTM + Attention. We initiated training for each dataset, however rather than training from scratch, we used the weights of the models trained on another STS dataset. We compared these transfer learning results to the results we got from training the model from scratch. We conducted this transfer learning experiment only on the STS2017 and SICK datasets since the QUORA dataset is already large and transfer learning from a smaller dataset to a larger dataset is nonsensical.

Start Model	STS2017	SICK
$STS2017_{GRU}$	0.853	(+0.01)
$STS2017_{LSTM+Aten}$	0.830	(+0.01)
$SICK_{GRU}$	(+0.01)	0.838
$SICK_{LSTM+Aten}$	(+0.01)	0.827
$QUORA_{GRU}$	(-0.02)	(-0.02)
$QUORA_{LSTM+Aten}$	(-0.04)	(-0.04)

Table 4.4: Results for transfer learning with different variants of Siamese Neural Network. For each transfer learning experiment we show the difference between with and without transfer learning. Non-grey values are the results of the experiments without transfer learning which we showed in the previous section. For ease of visualisation we only report the Pearson correlation (ρ).

As can be seen in Table 4.4 some of the transfer learning experiments improved the results for the STS2017 and SICK datasets with both architectures. The results improved when we performed transfer learning from STS2017 ⇒ SICK and SICK ⇒ STS2017. This shows that transfer learning can improve the results in Siamese neural networks. However, when we performed transfer learning from QUORA ⇒ STS2017 and QUORA ⇒ SICK, the results did not improve. In fact, they decrease, despite QUORA being the largest STS dataset we experimented with. This finding is somewhat controversial as the general belief in the community is that transfer learning from a larger dataset will improve the result. In this case, we believe that this happened because the QUORA dataset is very different to the other two datasets, as discussed in Chapter 1. Despite QUORA having a large number of training instances, when performing transfer learning, the neural network finds it difficult to optimise the weights for STS2017 and SICK as they were already optimised for a very different dataset; QUORA. This leads to a decrease in the results. On the other hand, transfer learning between STS2017 and SICK improved the results for both datasets since they are similar in nature, as we discussed in Chapter 1.

Therefore, we can conclude that transfer learning can improve the results for Siamese neural networks in STS. However, the transfer learning dataset should be picked carefully taking the similarity of the two datasets into consideration, rather than only considering the size of the dataset.

4.2.2 Impact of Data Augmentation

As we observed earlier, the neural networks perform better when there are large number of training instances. Therefore, many approaches have been taken to increase the number of training instances. Usually, this has resulted in better performance with neural networks (Wei and Zou 2019). Therefore, we examined the impact of data augmentation on the Siamese neural network architectures proposed previously. We only conducted this experiment with the STS 2017 and SICK datasets as QUORA already has a large number of training instances.

We employed thesaurus-based augmentation in which 10,000 additional training examples are generated by replacing random words with one of their synonyms from Wordnet (Miller 1995). A similar approach has also been successfully adapted by Mueller and Thyagarajan (2016), and Zhang et al. (2015). We specifically used the two models that performed best with the bigger dataset and smaller dataset; Siamese neural network with GRU and Siamese neural network with LSTM + Attention. Since using transfer learning improved the results in the previous experiment, we trained the augmented training set on the transferred models; models trained on STS2017 for the SICK experiments and models trained on SICK for the STS2017 experiments. The results are shown in Table ??.

Dataset	**Start Model**	ρ
SICK	$STS2017_{GRU}$	(+0.01)
	$STS2017_{LSTM+Aten}$	(+0.01)
STS2017	$SICK_{GRU}$	(+0.01)
	$SICK_{LSTM+Aten}$	(+0.01)

Table 4.5: Results for data augmentation with different variants of Siamese neural networks. For each data augmentation experiment, we show the difference between performing the data augmentation and without performing data augmentation. For ease of visualisation we only report the Pearson correlation (ρ).

As can be seen in Table ??, data augmentation improved the results of all the experiments. However, even with the additional 10,000 training instances, the GRU based Siamese neural network outperformed the LSTM + Attention based Siamese neural network. We can conclude that simple data augmentation techniques improve the performance of Siamese neural networks in STS tasks. From the Siamese neural network experiments we conducted, our best results for both the STS2017 and SICK datasets were provided by GRU based Siamese neural network when combined with transfer learning and data augmentation.

These observations answer our *RQ2* in this Chapter; we can use transfer learning and simple data augmentation techniques to improve the results of Siamese neural networks in STS.

Model	ρ
Jimenez et al. 2014	0.807
Bjerva et al. 2014	0.827
Zhao et al. 2014	0.841
Siamese LSTM	0.863
Siamese GRU	0.882

Table 4.6: Results for SICK dataset with different variants of Siamese neural networks. For each variant, Pearson Correlation (ρ) is reported between the predicted values and the gold labels of the test set.

Model	ρ
Tian et al. 2017	0.851
Siamese LSTM	0.852
Maharjan et al. 2017	0.854
Cer et al. 2017	0.855
Siamese GRU	0.862

Table 4.7: Results for STS2017 dataset with different variants of Siamese neural networks. For each variant, Pearson Correlation (ρ) is reported between the predicted values and the gold labels of the test set.

Furthermore, we compared the results of the best Siamese neural network variant with the best results submitted to the competitions (Cer et al. 2017; Marelli et al. 2014), and with the unsupervised STS methods we have experimented with so far in the thesis. As can be seen in Tables 4.6 and 4.7, the

GRU based Siamese neural network architecture outperforms the best systems submitted to both competitions. It also outperforms the unsupervised STS methods we have explored so far in the thesis. Therefore, we can conclude that Siamese architecture is currently the best system we have experimented with for English STS.

4.3 Portability to Other Languages

Our *RQ3* targets the multilingual aspect of the proposed approach; *Can the proposed Siamese neural network be easily adopted into different languages?*. To answer this, we evaluated our method in the Arabic STS and Spanish STS datasets that were introduced in Chapter 1. Our approach has the advantage that it does not rely on language-dependent features. As a result, the approach is easily portable to other languages, given the availability of pre-trained word embedding models in that particular language. The word embedding models, we used are AraVec (Soliman et al. 2017) [3] for Arabic and Spanish 3B words Word2Vec Embeddings (Bilbao-Jayo and Almeida 2018)[4] for Spanish.

As can be seen in Tables 4.8 and 4.9, the GRU based Siamese neural network outperformed all other variants we experimented with in both Arabic and Spanish. As we discussed in Chapter 1, both of the Arabic and Spanish STS datasets we considered are small in size. Therefore, similarly to the STS2017

[3] AraVec has been trained on Arabic Wikipedia articles. The models are available on https://github.com/bakrianoo/aravec

[4] Spanish 3B words Word2Vec Embeddings have been trained on Spanish news articles, Wikipedia articles and Spanish Boletín Oficial del Estado (BOE; English: Official State Gazette). The model is available on https://github.com/aitoralmeida/spanish_word2vec

Model	ρ	τ
LSTM	0.746	0.690
Bi-LSTM	0.725	0.683
GRU	$0.763^{\dagger}$	$0.723^{\dagger}$
Bi-GRU	0.752	0.717
LSTM + Attention	0.741	0.703
GRU + Attention	0.739	0.691
GRU + Capsule + Flatten	0.712	0.679

Table 4.8: Results for Arabic STS dataset with different variants of Siamese Neural Network. For each variant, Pearson Correlation (ρ) and Spearman Correlation (τ) are reported between the predicted values and the gold labels of the test set. The best result from all the variations is marked with †.

Model	ρ	τ
LSTM	0.842	0.773
Bi-LSTM	0.814	0.782
GRU	$0.863^{\dagger}$	$0.822^{\dagger}$
Bi-GRU	0.851	0.813
LSTM + Attention	0.845	0.801
GRU + Attention	0.832	0.790
GRU + Capsule + Flatten	0.795	0.773

Table 4.9: Results for Spanish STS dataset with different variants of Siamese Neural Network. For each variant, Pearson Correlation (ρ) and Spearman Correlation (τ) are reported between the predicted values and the gold labels of the test set. The best result from all the variations is marked with †.

and SICK datasets, the GRU based Siamese neural network outperforms other architectures as GRU does not need a lot of training instances to optimise its weights. It should be noted that it is very easy to adapt this STS method in a different language. We only changed the embeddings to the new language and then performed the training.

Furthermore, we compared the results of the best Siamese neural network variant with the best results submitted to the competition (Cer et al. 2017), and with the unsupervised STS methods we have experimented with so far in the thesis.

Model	ρ
Tian et al. 2017	0.744
Nagoudi et al. 2017	0.746
Siamese LSTM	0.746
Wu et al. 2017	0.754
Siamese GRU	0.763

Table 4.10: Results for Arabic STS dataset with different variants of Siamese Neural Network. For each variant, Pearson Correlation (ρ) is reported between the predicted values and the gold labels of the test set.

Model	ρ
Siamese LSTM	0.842
Hassan et al. 2017	0.848
Wu et al. 2017	0.850
Tian et al. 2017	0.855
Siamese GRU	0.863

Table 4.11: Results for Spanish STS dataset with different variants of Siamese Neural Network. For each variant, Pearson Correlation (ρ) is reported between the predicted values and the gold labels of the test set.

As can be seen in the results, transformer-based STS methods outperformed all the other supervised and unsupervised STS models in both languages. They outperformed the top systems from the competition in both languages. From the experimented pre-trained transformer models, language-specific models such as BETO and AraBERT outperformed the general multilingual models. With these observations, we can conclude that transformers are currently state-of-the-art in Arabic and Spanish STS. Furthermore, it should be noted that it is straightforward to adapt transformers to a different language. We only changed the pre-trained model to the new language and performed the training.

These observations answer our **RQ3**; the Siamese architectures that we propose in this chapter can be successfully adapted in different languages by changing the word embeddings and the training dataset.

4.4 Portability to Other Domains

To answer our *RQ4*; how well the proposed Siamese neural network architecture can be applied in different domains, we evaluated our method on the Bio-medical STS dataset explained in Chapter 1 (BIOSSES). As we mentioned previously, the Bio-medical STS dataset does not have a training set. Therefore, we had to follow a transfer learning strategy to evaluate Siamese neural networks on the Bio-medical STS dataset. We used the pre-trained English STS models and performed inference on the Bio-medical STS dataset. We can refer to this as a *"zero-shot transfer learning"* since the pre-trained English STS models did not see any Bio-medical data.

For this transfer learning strategy, we considered two word embedding models; the general Word2vec model we used before Mikolov et al. 2013a that was pre-trained on Google news corpus, and BioWordVec Zhang et al. 2019c, which has trained Word2vec on a combination of PubMed and PMC texts[5]. With each word embedding model, we trained a Siamese neural network based on GRU and a Siamese neural network based on LSTM + Attention (the two best models we had from the English STS experiments) and evaluated them on the BIOSSES

[5] The model is availble on `https://bio.nlplab.org/`

dataset.

Data	**Model**	**Word2vec**	**BioWordVec**
STS2017	*Siamese GRU*	0.651	0.721
	Siamese LSTM+Atten	0.612	0.701
SICK	*Siamese GRU*	0.642	0.719
	Siamese LSTM+Atten	0.608	0.699
QUORA	*Siamese GRU*	0.591	0.622
	Siamese LSTM+Atten	0.603	0.634

Table 4.12: Results for transfer learning with different variants of Siamese neural networks in BIOSSES dataset. **Data** column shows the datasets we performed transfer learning from and **Model** column displays the Siamese variant we employed. Two considered word embedding models are **Word2vec** and **BioWordVec**. For ease of visualisation we only report the Pearson correlation (ρ).

As you can see in Table 4.12, Siamese neural architectures provided satisfactory results. We got the best result from the Siamese neural network based on GRU, when trained on STS 2017 using BioWordVec. However, the results from the SICK dataset are not far behind. There was a clear improvement when the English STS model was trained using BioWordVec rather than general Word2vec embeddings. This may be because most of the Bio-medical words that appear in the BIOSSES dataset are out of vocabulary in general Word2vec embeddings, which can cause problems for the neural network when it observes them in the testing phase. It should be noted that in this experiment, when we performed transfer learning from the QUORA dataset, the results are lower than when we performed transfer learning from SICK or STS 2017. This again may be due to the fact that the SICK and STS2017 datasets have a similar annotation strategy to the BIOSSES dataset as discussed in Chapter 1. Even though QUORA has a large number of training instances, it can't produce good transfer learning results because its annotation strategy is different.

Model	ρ
ELMo $\bigoplus$ *BERT*	0.708
Siamese GRU$_{STS2017}$	0.719
Soğancıoğlu et al. 2017	0.754
BioSentVec Chen et al. 2019	0.810

Table 4.13: Results for BIOSSES dataset with different variants of Siamese Neural Network compared with top results reported for BIOSSES. For each variant, Pearson Correlation (ρ) is reported between the predicted values and the gold labels of the test set.

Furthermore, we compared our results with the best results reported for the dataset. The results are shown in Table 4.13. The best model we had in Table 4.12 which is the Siamese GRU model trained on the STS2017 dataset, is represented as *Siamese GRU*$_{STS2017}$. As shown in Table 4.13, our method provides satisfactory results compared with the best approaches submitted to the BIOESS dataset. However, the unsupervised method we experimented with in the previous chapter with BioSentVec (Chen et al. 2019), comfortably outperformed the Siamese neural network approaches we explored in this chapter. We can answer our **RQ4**: *How well does the proposed Siamese neural network perform in a different domain?* with these findings. The Siamese neural network architectures can be adapted to different domains by changing the pre-trained word embeddings. However, without a proper training set, the results are not strong.

4.5 Conclusions

This chapter experimented with using Siamese neural networks for calculating semantic similarity between pairs of texts and compared them with other unsupervised/ supervised approaches. We used an existing Siamese neural network as the baseline; MALSTM (Mueller and Thyagarajan 2016) and explored six different variants of Siamese neural networks. We experimented with three English STS datasets, SICK, STS2017 and QUORA. For the smaller STS datasets; SICK and STS2017, we show that the Siamese neural network based on GRU outperforms the baseline. For the larger STS dataset; QUORA, we show that Siamese neural network with LSTM and Attention outperforms the baseline. Also, we show that we can further improve the results with transfer learning and data augmentation techniques. However, we experienced that performing transfer learning from a bigger dataset does not always improve the results. The quality of the dataset which was used for transfer learning also matters. We show that Siamese neural network based on GRU outperforms the top submissions in both *SemEval 2017 task 1* (Cer et al. 2017) and *SemEval 2014 task 1* (Marelli et al. 2014). The data augmentation techniques we used in this chapter are language-dependent as they rely on WordNet (Miller 1995). However, as future work, we can experiment with data augmentation techniques that are not language dependant and relies on word embeddings (Kumar et al. 2020b).

We extended the experiments with the Siamese neural network architectures to the Arabic and Spanish STS datasets (Cer et al. 2017). In these experiments, the GRU based Siamese neural network architecture again outperformed all the systems submitted to the shared task and outperformed all of the STS methods we have experimented with so far in this part of the thesis. This proves that the Siamese neural network that we propose in this study, can be adapted to different languages. Furthermore, we performed experiments with the BIOSSES dataset. However, since the BIOSSES dataset does not have a training set, we used transfer learning based zero-shot learning when Siamese neural networks are applied. Even though they provided satisfactory results, Siamese neural networks could not outperform the sentence encoder based method we explored in Chapter 3. We can conclude that even though the Siamese neural networks can be adapted into different domains by changing the word embedding model, they do not provide strong results without a proper training set.

Since word embedding models are now available in most languages, including low resource languages such as Urdu (Haider 2018), Telugu (Kumar et al. 2020a) and domains such as the legal domain (Chalkidis and Kampas 2019), the method we experimented with in this chapter can be useful for many languages and domains. However, one drawback is the need for STS training data in each language and domain, this can be challenging in many scenarios.

As future work, it would be interesting to experiment transfer learning between languages with cross-lingual embeddings such as fastText (Mikolov et al. 2018) using Siamese neural networks. Such an approache will train an STS model on resource-rich languages like English and project the prediction for other languages using the zero-shot transfer learning. It would be a potential solution to satisfy the training data requirement for low resource languages.

With the introduction of transformer models such as BERT (Devlin et al. 2019) and XLNet (Yang et al. 2019b), Siamese neural networks have evolved by utilising transformers in their architectures (Reimers and Gurevych 2019). We will discuss these further in Chapter 5.