

# Logic Constrained Pointer Networks for Interpretable Textual Similarity

Subhadeep Maji<sup>1\*†</sup>, Rohan Kumar<sup>1\*</sup>, Manish Bansal<sup>1</sup>, Kalyani Roy<sup>2</sup> and Pawan Goyal<sup>2</sup>

<sup>1</sup>Flipkart

<sup>2</sup>Indian Institute of Technology, Kharagpur

msubhade@amazon.com, {rohankumar, manish.bansal}@flipkart.com,  
kroy@iitkgp.ac.in, pawang@cse.iitkgp.ac.in

## Abstract

Systematically discovering semantic relationships in text is an important and extensively studied area in Natural Language Processing, with various tasks such as entailment, semantic similarity, etc. Decomposability of sentence-level scores via subsequence alignments has been proposed as a way to make models more interpretable. We study the problem of aligning components of sentences leading to an interpretable model for semantic textual similarity. In this paper, we introduce a novel pointer network based model with a sentinel gating function to align constituent chunks, which are represented using BERT. We improve this base model with a loss function to equally penalize misalignments in both sentences, ensuring the alignments are bidirectional. Finally, to guide the network with structured external knowledge, we introduce first-order logic constraints based on ConceptNet and syntactic knowledge. The model achieves an F1 score of 97.73 and 96.32 on the benchmark SemEval datasets for the chunk alignment task, showing large improvements over the existing solutions. Source code is available at [https://github.com/manishb89/interpretable\\_sentence\\_similarity](https://github.com/manishb89/interpretable_sentence_similarity)

## 1 Introduction

Measuring semantic similarity between sentences has been one of the major problems towards text understanding. Many tasks including paraphrase identification [Socher *et al.*, 2011], text entailment recognition [Heilman and Smith, 2010], etc. also utilize sentence similarity. Clearly, it has attracted a lot of attention in the NLP research community [Shao, 2017; Tai *et al.*, 2015]. Semantic textual similarity (STS) dataset from SemEval 2012 [Agirre *et al.*, 2012] has been one of the commonly used benchmark for sentence similarity task, which attempts at measuring the degree of semantic equivalence between two sentences. While recently proposed deep learning methods built on pretrained language models have shown great success for the task [Reimers and

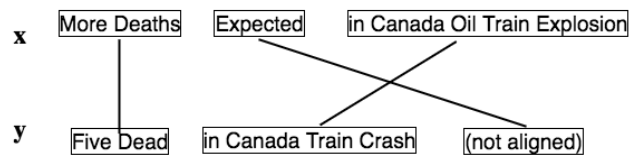


Figure 1: Interpretable sentence similarity defined as alignments between chunks (indicated by connecting lines) of the two sentences. An unaligned chunk is considered aligned to special ‘not aligned’ chunk (denoted by  $\phi$  in paper).

Gurevych, 2019], interpretability and explainability of the final scores remains a concern in general. [Agirre *et al.*, 2015] proposed to formalize interpretable semantic textual similarity (henceforth, iSTS) as an alignment between pairs of segments across the two sentences at SemEval 2016. This is also linguistically well motivated because similarity between sentences has been observed to be decomposable [Sultan *et al.*, 2015] over segments and the overall similarity is a combined measure of similarity on parts (words, chunks, etc). The problem of interpretable semantic textual similarity is to provide an interpretation or explanation of semantic similarity between two texts (usually sentences). We consider chunking to be a preprocessing step and assume that the sentences are already chunked. Not all chunks in the sentences may be aligned and number of aligned chunks gives indication to the overall similarity between the sentences. Figure 1 provides an illustrative example.

We introduce a novel logic statement constrained gated pointer network model to align constituents of the two sentences, aiding in interpretation of semantic relationships between sentences. Our model uses a pointer network [Vinyals *et al.*, 2015] with a sentinel gating function to align the constituent chunks, which are represented using BERT. We improve this base model with a loss function to equally penalize misalignments in both sentences, ensuring bidirectional alignments. Finally, we introduce first-order logic constraints based on ConceptNet as well as syntactic knowledge to aid in training and inference.

Experiments over two different SemEval datasets indicate that our proposed approach achieves state of the art results on both the datasets. We achieve F1 score of 97.73 on headlines and 96.32 on images, an improvement of 7.8% and 5.6% over

\*Equal contribution.

†Now at Amazon.

the previous best results, respectively. Through ablation studies, we also find that the proposed logical constraints help boost the performance on both the datasets. Further, since getting alignments for training the model is costly, we also perform a cross-domain experiment (training on headlines, testing on images and vice-versa) and find that even in this scenario, we achieve F1 scores of 96.16 and 94.80 on these datasets, comprehensively beating state-of-the-art methods.

## 2 Related Work

Discovering semantic relations in text is a widely studied problem in Natural Language Processing (NLP) with various tasks. One such task is STS which was first introduced as a shared task in [Agirre *et al.*, 2012]. Several approaches to this task have been developed [Bär *et al.*, 2012; Jimenez *et al.*, 2012], with deep learning based methods achieving the most success recently [Shao, 2017; Tai *et al.*, 2015; Reimers and Gurevych, 2019]. [Shao, 2017] train a CNN-based sentence encoder on the STS task. [Conneau *et al.*, 2017] and [Pagliardini *et al.*, 2018] train sentence encoders on alternative similar tasks to aid in learning. Many such approaches make use of an alignment (implicitly or explicitly) between parts of sentences (e.g. [Hänig *et al.*, 2015; Sultan *et al.*, 2015]) with the assumption that composing subsequence alignments can lead to better identification of semantic similarity. Alignments also lend a notion of interpretability of sentence level similarity judgements.

Alignment of parts of sentence pairs is also well studied in domains such as Machine Translation [Koehn *et al.*, 2003] and Paraphrase Recognition [Chang *et al.*, 2010]. The iSTS [Agirre *et al.*, 2015] shared task focuses on predicting chunk alignments as a means to lend interpretability to STS. [Banjade *et al.*, 2015] develop a rule-based alignment system using standard textual features such as parts-of-speech (POS) tags. [Kazmi and Schüller, 2016] realize a similar rule engine in Answer Set Programming, allowing reordering of rules. [Konopik *et al.*, 2016] pose chunk alignment as a binary classification task supported with rules. [Li and Srikumar, 2016] build an alignment model using an integer linear program (ILP), with scoring functions learnt using both alignment and sentence similarity tasks. Chunk alignment is a central piece in this line of work and is also the focus of our work.

Incorporating external information from large labelled or unlabelled corpora into neural models for NLP has been shown to improve task performance. One common way is to use pre-trained word or sentence embeddings. In this work, we experiment with GloVe [Pennington *et al.*, 2014] and BERT [Devlin *et al.*, 2019] based chunk representations and show ablations in model performance. Another external source of information is in the form of structured knowledge from knowledge bases like ConceptNet [Speer *et al.*, 2017] and Paraphrase Database (PPDB) [Pavlick *et al.*, 2015]. Structured knowledge can be incorporated in the form of first-order logic statements without expensive direct supervision. Logic rules provide a declarative language to express such structured knowledge and thus have been used to guide neural networks in several ways recently. [Towell *et al.*, 1990] and [França *et al.*, 2014] build networks from a rule set expressed

in propositional logic. [Hu *et al.*, 2016] make use of teacher-student network formulation to distill logic rules into network parameters. This approach does not restrict models to specialized networks but requires a specialized training procedure. [Li and Srikumar, 2019] introduce logic statements to constraint labelled neurons (e.g. attention nodes) which can be used in general neural networks. We introduce constraints in the framework which are applicable to the chunk alignment task.

## 3 Approach

Given two sentences, let us denote these by  $\mathbf{x} = (x_1, x_2, \dots, x_n)$  and  $\mathbf{y} = (y_1, y_2, \dots, y_m)$ , where  $x_i$  and  $y_j$  are chunks (i.e contiguous words) in  $\mathbf{x}$  and  $\mathbf{y}$ , respectively. The problem of iSTS is to predict an alignment between chunks  $z_{i,j} \in \{0, 1\}$  indicating if  $x_i$  and  $y_j$  are aligned. We consider a supervised setting in which the ground truth alignment  $a_{i,j} \in \{0, 1\}$  is specified in the training data. Not all chunks in  $\mathbf{x}$  are aligned to a chunk in  $\mathbf{y}$  (and vice-versa). We consider these non-aligned chunks to be aligned to a special  $\phi$  chunk. Therefore, the problem is to generate an alignment from  $\mathbf{x}$  to  $\mathbf{y} \cup \{\phi\}$  and from  $\mathbf{y}$  to  $\mathbf{x} \cup \{\phi\}$ . Figure 2 shows a block level view of our model. Next, we explain each of the individual components in further details.

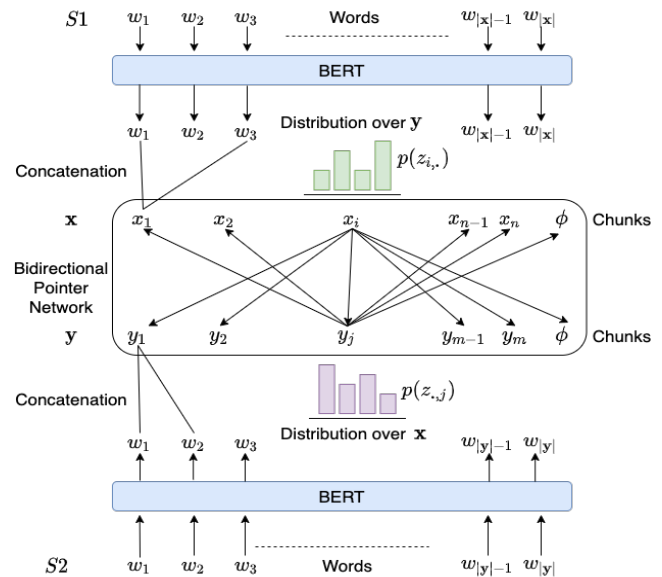


Figure 2: Block level illustration of our Bidirectional Pointer Network with BERT based chunk embeddings. For ease of illustration, we show the non-aligned chunk  $\phi$  on both sentences and do not show FOL constraints.

### 3.1 Chunk Representation

Given a chunked sentence, we obtain chunk representation from BERT [Devlin *et al.*, 2019] *bert-base-uncased* variant by concatenation of contextualized embedding of first and last word of a chunk. We use summation of last 4-layers from BERT to represent a word as higher layers of BERT capture

semantic features as suggested in [Jawahar *et al.*, 2019]. We use BERT WordPiece model for sub-words representations corresponding to a word which are averaged to realize the word final representation. We also experiment with GloVe where the chunk representation is obtained by averaging the word embeddings, as is common in literature.

### 3.2 Alignment with Gated Pointer Networks

Extending previous work [Vinyals *et al.*, 2015], we design a pointer network (PN) to model alignment between chunks of sentences  $\mathbf{x}$  and  $\mathbf{y}$ . The alignment between chunks  $x_i$  and  $y_j$  is modelled as

$$\theta_{i,j} = v^T f(W_1 x_i + W_2 y_j + W_3 x_i \otimes y_j) \quad (1)$$

where  $W_1, W_2, W_3$  and  $v$  are model parameters,  $f$  is a non-linearity (we use  $\tanh$ ) and  $\otimes$  is the Hadamard product. The  $W$  matrices are of the same dimension and project chunk embeddings to a lower dimension  $d$ ;  $v$  is a  $d$ -dimension vector. Thus, the PN ‘points’ from chunks in  $\mathbf{x}$  to chunks in  $\mathbf{y}$ . However, we consider the pointers as unidirectional and interpret Equation (1) as alignment between  $y_j$  and  $x_i$  as well. To allow for non-aligned chunks, we consider such chunks to be aligned to a special empty chunk  $\phi$  and treat its representation as a parameter of the model. Specifically with a little abuse of notation, we model non-aligned chunks  $x_i$  and  $y_i$  using  $b_{i,\phi}^x$  and  $b_{\phi,i}^y$ , respectively, given by

$$b_{i,\phi}^x = v^T f(W_1 x_i + W_2 \phi) \quad (2)$$

$$b_{\phi,i}^y = v^T f(W_1 \phi + W_2 y_i)$$

Using Equations (1) and (2), we define a gating function that models  $x_i$ ’s alignment to  $\mathbf{y}$  and  $y_i$ ’s alignment to  $\mathbf{x}$ ,

$$g_i^x = \sigma(c_1 \max_j (\theta_{i,j} - b_{i,\phi}^x) + c_2) \quad (3)$$

$$g_j^y = \sigma(d_1 \max_j (\theta_{j,i} - b_{\phi,i}^y) + d_2)$$

where  $c_1, c_2$  and  $d_1, d_2$  are parameters of the model. Intuitively,  $g_i^x$  captures how well  $x_i$  is aligned to its best alignment (with high  $\theta_{i,j}$ ) in  $\mathbf{y}$  in comparison to  $\phi$ . Mathematically,  $g_i^x \rightarrow 1$  when  $x_i$  is better aligned to some  $y_j$  in comparison to  $\phi$ . Let  $z_{i,j} \in \{0, 1\}$  indicate if  $x_i$  is aligned to  $y_j$  ( $z_{i,j} = 1$ ). Using Equations (3) and (1), we model the probability of the event  $z_{i,j}$  as,

$$p(z_{i,j} = 1) \propto g_i^x g_j^y \theta_{i,j} \quad (4)$$

Thus, the alignment probability of  $x_i$  and  $y_j$  is proportional to product of  $x_i$  being aligned to some chunk in  $\mathbf{y}$ ,  $y_j$  being aligned to some chunk in  $\mathbf{x}$  and  $x_i$  being aligned to  $y_j$ .

The probability of non-alignment of  $x_i$  (aligned to  $\phi$ ) is proportional to  $1 - g_i^x$  and probability of non-alignment of  $y_j$  is proportional to  $1 - g_j^y$ . Thus

$$p(z_{i,\phi} = 1) \propto (1 - g_i^x) \quad (5)$$

$$p(z_{\phi,j} = 1) \propto (1 - g_j^y)$$

Equations (4) and (5) are appropriately normalised by passing them through a softmax layer. Mathematically,

$$\sum_{j=1}^m p(z_{i,j} = 1) + p(z_{i,\phi} = 1) = 1 \quad (6)$$

$$p(z_{i,j} = 1) = \text{softmax}([g_i^x g_j^y \theta_{i,j}; (1 - g_i^x)])$$

where  $[\cdot; \cdot]$  indicates concatenation.

Therefore, the alignment is modeled for every index  $i$  in  $\mathbf{x}$  because  $p(z_{i,\cdot})$  is a distribution for every  $i$ . The model is trained by minimizing the categorical cross-entropy loss between correct ( $a_{i,j}$ ) and predicted alignment ( $z_{i,j}$ ). However, this loss alone fails to capture the non-alignment for chunks in  $\mathbf{y}$  because the PN ‘points’ from  $\mathbf{x}$  to  $\mathbf{y}$  and therefore cannot model  $\phi$  aligning to multiple  $y_j$ ’s. We address this separately by binary cross-entropy loss between  $p_{\phi,j}$  and non-aligned chunks in  $\mathbf{y}$  indicated by  $a_{\phi,j}$ . The combined loss is addition of these two losses,

$$- \frac{C_1}{n} \sum_{i=1}^n \left( \sum_{j=1}^m a_{i,j} \log(p_{i,j}) + a_{i,\phi} \log(p_{i,\phi}) \right) \quad (7)$$

$$+ C_2 \left( - \sum_{i=1}^n a_{\phi,i} \log(p_{\phi,i}) + (1 - a_{\phi,i}) \log(1 - p_{\phi,i}) \right)$$

where  $C_1, C_2$  are positive hyperparameters capturing the relative cost sensitivity between the two loss functions.

### 3.3 Improving Alignments with Bidirectionality

For a pair of similar sentences, usually a certain part of a sentence is semantically similar to only a particular part of the other sentence. Thus, the alignment is usually one-to-one between chunks in the two sentences. We observed this phenomenon in SemEval 2016 dataset on iSTS, where approximately 85% of alignments between the chunks in the sentence pair are one-to-one. However, the gated PN formulation we developed in Section 3.2 falls short in modelling one-to-one alignments. This is because while alignment for every chunk  $x_i$  in  $\mathbf{x}$  is modelled explicitly as a distribution  $z_{i,\cdot}$ , thus encouraging a hard alignment to a chunk  $y_j$  (because of the cross-entropy loss), the alignments for  $y_j$  are unconstrained (e.g the model might easily choose to align two chunks  $x_i$  and  $x_k$  to  $y_j$ ). Therefore, the one-to-one alignment for  $y_j$ ’s is violated because it is not being modelled, and is implicitly derived from  $z_{\cdot,j}$ ’s. We illustrate this in Figure 3 with an example from the SemEval dataset.

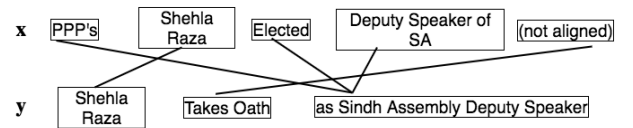


Figure 3: Alignment as obtained by gated PN in Section 3.2. Many-to-one alignment obtained by the model for a chunk in  $\mathbf{y}$  is incorrect. This is because it only models  $\mathbf{x}$  to  $\mathbf{y}$  alignment as a distribution.

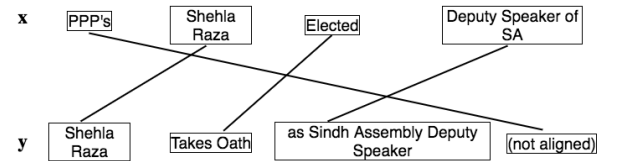


Figure 4: Enforcing bidirectionality on gated PN improves the alignment. The alignment is one-to-one and corresponds to the ground truth alignment.

We address this shortcoming of our gated PN by ensuring bidirectionality, i.e., both  $z_{i,\cdot}$  and  $z_{\cdot,j}$  are distributions for all  $i, j$ . Moreover,  $x_i$ 's alignment to  $y_j$  and  $y_j$ 's alignment to  $x_i$  is modeled by the same event  $z_{i,j}$  with probability  $p_{i,j}$ . Alignment between two sentences can be viewed as solution to an optimal transport problem where the transportation cost is defined by the activations in Equations (4) and (5). This results in a combinatorial optimization and is difficult to solve in an end-to-end manner with a neural network.

We propose an approximate solution to this problem using Sinkhorn distance [Cuturi, 2013] which is an entropy regularized approximation of optimal transportation problem. Mathematically, Sinkhorn distance is a solution  $\mathbf{p}$  to,

$$\begin{aligned} \min_{\mathbf{p}} \quad & \sum_{i,j \in \{\phi\}} p_{i,j} C_{i,j} - \lambda H(\mathbf{p}) \quad (8) \\ \text{s.t.} \quad & \forall i \sum_{j \in \{\phi\}} p_{i,j} = 1 \quad \forall j \sum_{i \in \{\phi\}} p_{i,j} = 1 \end{aligned}$$

where  $\mathbf{p}$  is a distribution,  $H(\mathbf{p})$  is the entropy of  $\mathbf{p}$ ,  $\mathbf{C}$  is a transportation cost matrix and  $\lambda$  is the strength of the entropy regularization. The constraints in Equation (8) ensure that  $\mathbf{p}$  is a distribution over alignments from  $\mathbf{x}$  to  $\mathbf{y}$  and from  $\mathbf{y}$  to  $\mathbf{x}$ . We define the cost matrix  $\mathbf{C}$  as,

$$\begin{aligned} C_{i,j} &= -g_i^x g_j^y \theta_{i,j} \\ C_{i,\phi} &= -(1 - g_i^x) \\ C_{\phi,j} &= -(1 - g_j^y) \end{aligned}$$

In transportation problems, the cost matrix is defined as non-negative. To ensure non-negativity of the cost matrix we subtract the minimum of each row from the entries of the matrix to define,  $C_{i,j} = C_{i,j} - \min_j C_{i,j} \quad \forall i$ . We use Cuturi's iterative row and column normalization algorithm [Cuturi, 2013] to solve Equation (8). To ensure convergence of Cuturi's algorithm [Cuturi, 2013], a small constant  $\epsilon$  is added to every entry of the cost matrix. The parameters of the model are learnt in an end-to-end manner. The backward pass to compute gradients differentiates through Cuturi's algorithm. This is achievable with automatic differentiation because the algorithm consists of only multiplication and division steps over differentiable terms.

Now both  $p_{i,\cdot}$  and  $p_{\cdot,j}$  form distributions and the modified formulation can be viewed as a bidirectional PN which points from both  $\mathbf{x}$  to  $\mathbf{y}$  and from  $\mathbf{y}$  to  $\mathbf{x}$ . The loss function is categorical cross-entropy on  $\mathbf{p}$ . The two-way alignment distribution encourages one-to-one alignments for chunks in both  $\mathbf{x}$  and  $\mathbf{y}$ ,

$$\begin{aligned} 2 \left( - \sum_{i=1}^n \sum_{j=1}^m a_{i,j} \log(p_{i,j}) \right) \quad (9) \\ + \left( - \sum_{i=1}^n a_{i,\phi} \log(p_{i,\phi}) \right) + \left( - \sum_{j=1}^m a_{\phi,j} \log(p_{\phi,j}) \right) \end{aligned}$$

Equation (9) can be viewed as summation of categorical cross-entropy of distributions  $p_{i,\cdot}$  and  $p_{\cdot,j}$ . Figure 4 illustrates the improvement in alignment on the example given in Figure 3.

### 3.4 Side-supervision with FOL

Guiding neural networks with structured external knowledge has been shown to improve predictive performance by complementing powerful data-driven learning. This knowledge

provides a mechanism to regulate the learning process by encoding human intention without expensive supervision. A number of techniques have been proposed to incorporate declarative logic statements into networks, such as data augmentation [Collobert *et al.*, 2011], knowledge distillation [Hu *et al.*, 2016], etc. [Li and Srikumar, 2019] propose a promising technique to augment existing networks by constraining activations of named neurons (e.g. attention layer) using declarative rules. Motivated by their approach, in this work, we propose two intuitive rules for the chunk alignment task,

- **R1**: Two chunks should be aligned if they are related by any of the relations: Synonym, Antonym, IsA, SimilarTo, RelatedTo, DistinctFrom or FormOf. The relational information to realize this rule is obtained from ConceptNet.
- **R2**: Two chunks should be aligned if they are syntactically similar. This similarity is defined based on overlap of parts-of-speech tags of their ancestor or children nodes in dependency parse trees of the two sentences.

These rules are based on structured knowledge and are expressed as declarative statements. We define predicates  $Rel_{i,j}$  and  $SynSim_{i,j}$  to indicate whether chunk  $x_i$  is related to chunk  $y_j$  according to **R1** or **R2**, respectively. Let  $A_{i,j}$  denote the model decision that chunk  $x_i$  is related to chunk  $y_j$ . We use the following constraints:

$$\begin{aligned} \forall i, j \in W, Rel_{i,j} \rightarrow A_{i,j} \quad (10) \\ \forall i, j \in W, SynSim_{i,j} \rightarrow A_{i,j} \end{aligned}$$

For rule **R1**, we experiment with two resources, ConceptNet and the paraphrase database (PPDB). In the ConceptNet knowledge base permitted relation types are [Synonym, Antonym, IsA, SimilarTo, RelatedTo, DistinctFrom, FormOf] which closely correspond to the alignment categories specified in the task [Agirre *et al.*, 2015]. We also experiment with PPDB since a large number of alignments are paraphrases. As complete chunks may not always be available, we evaluate using bigram and unigram alignments from these sources as indicators of chunk alignments. We ignore non-content words by restricting the alignment to content words  $W$ , identified as having POS tags [ADJ, ADV, INTJ, NOUN, PROP, VERB, NUM].

We next discuss the details of syntactic similarity measure which constitutes **R2**. Using the dependency tree of  $\mathbf{x}$  and  $\mathbf{y}$  obtained from Spacy [Honnibal and Montani, 2017], we define syntactic similarity measure between two words  $w_1 \in x_i$  and  $w_2 \in y_j$  as an average of (i). Jaccard similarity between parts-of-speech (POS) tags of ancestor nodes of  $w_1$  and  $w_2$ , (ii). Jaccard similarity between POS tags of children nodes of  $w_1$  and  $w_2$  and (iii). Boolean to indicate if both  $w_1$  and  $w_2$  are roots of the corresponding dependency trees. The syntactic similarity between  $x_i$  and  $y_j$  is defined as the average syntactic similarity between words in  $x_i$  to its best aligned word (maximum similarity) in  $y_j$ . A high value of syntactic similarity between  $x_i, y_j$  indicates a possible alignment.

Using the logic statements from Equation (10) we constraint the pointer network decisions by adding a positive constant  $m_{i,j}$  to the activations in Equation (1) if either **R1** or **R2** is true on  $(i, j)$  (i.e. a rule aligns  $x_i$  and  $y_j$ ),

$$\theta'_{i,j} = \theta_{i,j} + \rho m_{i,j} \quad (11)$$

where  $m_{i,j}$  captures whether the antecedent  $Rel_{i,j}$  or  $SynSim_{i,j}$  or both are true. From an implementation perspective, we unroll the FOL statement to propositional statements for all examples. The modified alignment strength  $\theta'$  in Equation (11) replaces  $\theta$  and affects both the gating functions in Equation (3) and combined alignment probability in Equation (4). This saturates the probability of the event  $z_{i,j}$  to 1 constraining the network output towards rule predictions. The importance of FOL rules is controlled using the  $\rho$  hyperparameter.

Hyperparameter	Range
$\rho$	[0, 1, 2, 4]
PN dimension( $d$ )	[100, 150, 200, 768]

Table 1: Hyperparameter configurations. The  $\rho$  value of 0 indicates a model without constraints. The PN dimension of 768 is valid only for BERT chunk based representation models (M3 & M4).

## 4 Experiments

We compare the proposed model against existing work on iSTS. In line with much of previous research on this task, we report experimental results on the SemEval 2016 interpretable textual similarity dataset.

### 4.1 Dataset Description

We use SemEval 2016 Task 2 dataset for interpretable semantic textual similarity [Agirre *et al.*, 2016]. It consists of examples from two domains; News Headlines and Flickr Images. In both domains, there are 1,125 sentence pairs, with a 2:1 split between train and test sets. Each sentence pair is annotated with the alignments between the chunks, similar to example in Figure 1. In this work, we focus on predicting the alignment between the chunks. Additionally, the chunking of the sentences is pre-specified in both train and test examples, therefore we directly use the chunking in this work. For more detailed specifications, please refer to [Agirre *et al.*, 2016].

### 4.2 Baselines & Models

We compare against the best task submissions on the two datasets as well as a followup work achieving SOTA results. Inspire [Kazmi and Schüller, 2016] is a rule-based alignment system extending the earlier NeRoSim [Banjade *et al.*, 2015] rule engine. It introduces Answer Set Programming to build an extended rule set. This model was the winning entry of the SemEval task on the News Headlines dataset. UWB [Konopik *et al.*, 2016] pose alignment as binary classification between all possible chunk pairs using lexical, syntactic, semantic and WordNet-based features, with impossible alignments handled via rules. This model was the winning entry of the SemEval task on the Flickr Images dataset. Lastly, we compare against [Li and Srikumar, 2016] which models chunk alignment using an ILP. Scoring functions for the ILP are calculated using a structured loss to penalize alignments far-off from ground truth along with two additional terms for chunk and sentence similarities. To the best of our knowledge, currently this method achieves SOTA results for the iSTS task on the SemEval dataset.

Model	Configurations				Dataset	
	BERT	Glove	FOL Constraints	Bidirectional PN	Headlines	Images
Inspire	-	-	-	-	81.94	86.7
UWB	-	-	-	-	89.87	89.37
ILP	-	-	-	-	92.57	87.38
M1		✓	×	×	89.7	88.34
M2		✓	×	✓	91.48	90.88
M3	✓		×	✓	96.63	93.81
M4	✓		✓	✓	<b>97.73</b>	<b>96.32</b>

Table 2: Average F1 score of alignments on the test set. The results for the models M1 to M4 are on the best hyperparameter configuration for each. Our best model M4 is better than the existing SOTA method by 6.7% on average across the datasets.

To thoroughly investigate the proposed approach, we report the experimental results across four different configurations of the proposed model, as described in Table 2. These settings capture the relative merit of each of the model components. The evaluation metric is F1 measure as per the SemEval task description.

### 4.3 Evaluation

We report F1 measure across the model ablations and baselines discussed in Section 4.2. The results for the baselines are reported from the corresponding papers owing to unavailability of their code bases. We trained each of the model configurations on train part of the SemEval dataset and did hyperparameter tuning on training set F1. We fixed the entropy regularization strength  $\lambda$  to 0.6 across all experiments and changing it had little effect on results. The embedding dimension for Glove based representations was 300 and for BERT was 768. To avoid over-fitting, we employed early stopping using training set F1 as a metric and stop the training if training set F1 does not improve over 5 successive epochs. In Table 2, we report the average test set F1 over 3 runs for each model configurations (corresponding to the best hyperparameter setting) on news headlines and images dataset. For example, the best identified hyperparameter configuration corresponding to M4 are;  $\rho = 2$  and PN dimension of 100 for experiments on headlines dataset and  $\rho = 2$  and PN dimension of 150 for experiments on images dataset. The results for M4 and M3 in comparison to M2 and M1 indicate that BERT based chunk embeddings are superior to Glove based representation. The constraints improve the performance of the BERT based model as highlighted by superior performance of M4 over M3. The improvements using constraints are much more visible on the images dataset. Both BERT chunk representation based models are significantly better than SOTA results [Li and Srikumar, 2016; Konopik *et al.*, 2016]. We see an improvement of 5.6% on News headlines dataset and 7.8% on Flickr images dataset for M4 over SOTA results. The Glove representation based model even without FOL constraints (M2) is comparable to SOTA method on both the datasets.

### 4.4 Qualitative Evaluation

We illustrate the qualitative merits of different components of our model on two examples by investigating the alignment produced by the four modelling configurations we introduced

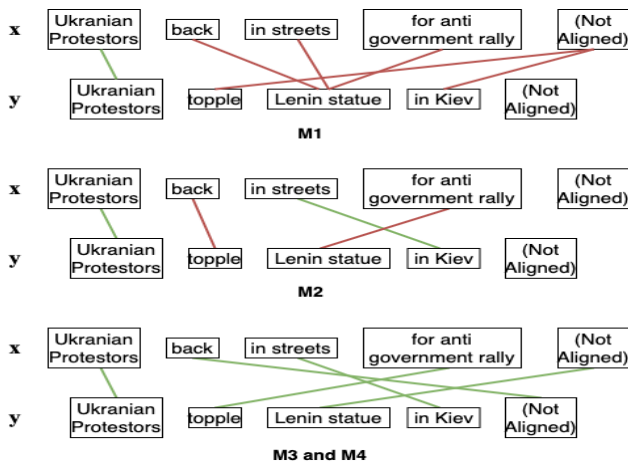


Figure 5: (Best viewed in color) Relative merit of M3 & M4 over M1 & M2. The correct alignments according to ground truth are shown in green and the incorrect ones are shown in red.

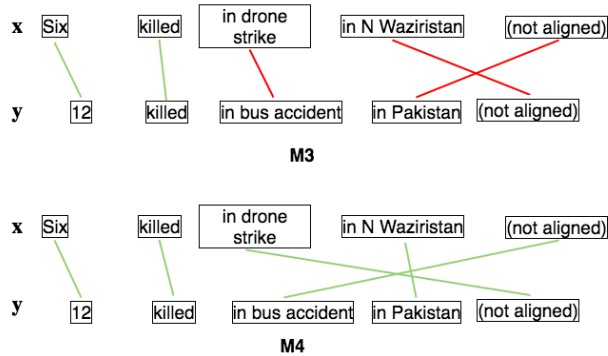


Figure 6: (Best viewed in color) Relative merit of M4 over M3. The correct alignments according to ground truth are shown in green and the incorrect ones are shown in red.

in Section 4.2. Figure 5 shows that modelling bidirectionality is important as is evident from large number of alignment errors made by M1 in comparison to M2. However, using BERT based contextual embeddings for chunks in the sentences leads to improved representations for chunks and both M3 and M4 correctly align all the chunks.

Figure 6 shows the effect of including real-world knowledge in the form of FOL statements in the model. The chunks pair “in N Waziristan” and “in Pakistan” are aligned by the FOL statements retrieved from ConceptNet. This is because ConceptNet holds the real-world knowledge that “Waziristan” is a part of “Pakistan” and relates them. BERT embeddings alone may not encode enough information to relate “Waziristan” and “Pakistan” and M3 fails to align both these chunks correctly, while M4 owing to the activated FOL constraint gives the correct alignment.

#### 4.5 Cross-domain Experiments

While the proposed approach achieves significant improvements over the previous baselines, a natural question arises – how would this approach be applicable on a new sentence

similarity dataset to provide interpretability, when no chunk alignments are available to train the model in the target domain? To answer this, we attempted a cross-domain experiment, where we train on headlines, test on images, and vice versa (i.e., do not utilize training examples from the target domain). Remarkably, even in this setting, using our best model M4, we achieve F1 scores of 96.16 and 94.80 on headlines and images datasets, respectively, outperforming the previous SOTA results. Note that no hyperparameter tuning was performed on the target domain.

## 5 Conclusion

We propose a novel pointer network for the task of interpretable sentence similarity along with logic constraints based on ConceptNet and syntactic knowledge. Experiments over benchmark datasets show a large performance improvement for the alignment task, even in the cross-domain setting, proving the general applicability of the proposed approach.

It was encouraging to see that the logical constraints imposed using external knowledge helped the model performance, and it would be interesting to check how the whole framework can be employed to improve performance for the sentence similarity task, while providing interpretability and explanation for the model decision.

## References

[Agirre *et al.*, 2012] Eneko Agirre, Mona Diab, Daniel Cer, and Aitor Gonzalez-Agirre. Semeval-2012 task 6: A pilot on semantic textual similarity. In *STARSEM and SemEval*, pages 385–393. Association for Computational Linguistics, 2012.

[Agirre *et al.*, 2015] Eneko Agirre, Carmen Banea, Claire Cardie, Daniel Cer, Mona Diab, Aitor Gonzalez-Agirre, Weiwei Guo, Inigo Lopez-Gazpio, Montse Maritxalar, Rada Mihalcea, et al. Semeval-2015 task 2: Semantic textual similarity, english, spanish and pilot on interpretability. In *SemEval*, pages 252–263, 2015.

[Agirre *et al.*, 2016] Eneko Agirre, Aitor Gonzalez-Agirre, Inigo Lopez-Gazpio, Montse Maritxalar, German Rigau, and Larraitz Uria. Semeval-2016 task 2: Interpretable semantic textual similarity. In *SemEval*, pages 512–524, 2016.

[Banjade *et al.*, 2015] Rajendra Banjade, Nopal Bikram Niraula, Nabin Maharjan, Vasile Rus, Dan Stefanescu, Mihai Lintean, and Dipesh Gautam. Nerosim: A system for measuring and interpreting semantic textual similarity. In *SemEval*, pages 164–171, 2015.

[Bär *et al.*, 2012] Daniel Bär, Chris Biemann, Iryna Gurevych, and Torsten Zesch. Ukp: Computing semantic textual similarity by combining multiple content similarity measures. In *STARSEM and SemEval*, pages 435–440. Association for Computational Linguistics, 2012.

[Chang *et al.*, 2010] Ming-Wei Chang, Dan Goldwasser, Dan Roth, and Vivek Srikumar. Discriminative learning over constrained latent representations. In *NAACL-HLT*, pages 429–437. Association for Computational Linguistics, 2010.

- [Collobert *et al.*, 2011] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch. *JMLR*, 12(Aug):2493–2537, 2011.
- [Conneau *et al.*, 2017] Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. Supervised learning of universal sentence representations from natural language inference data. In *EMNLP*, pages 670–680, 2017.
- [Cuturi, 2013] Marco Cuturi. Sinkhorn distances: Light-speed computation of optimal transport. In *NeurIPS*, pages 2292–2300, 2013.
- [Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, pages 4171–4186, 2019.
- [França *et al.*, 2014] Manoel VM França, Gerson Zaverucha, and Artur S d’Avila Garcez. Fast relational learning using bottom clause propositionalization with artificial neural networks. *Machine learning*, 94(1):81–104, 2014.
- [Hänig *et al.*, 2015] Christian Hänig, Robert Remus, and Xose De La Puente. Exb themis: Extensive feature extraction from word alignments for semantic textual similarity. In *SemEval*, pages 264–268, 2015.
- [Heilman and Smith, 2010] Michael Heilman and Noah A Smith. Tree edit models for recognizing textual entailments, paraphrases, and answers to questions. In *NAACL-HLT*, pages 1011–1019. Association for Computational Linguistics, 2010.
- [Honnibal and Montani, 2017] Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.
- [Hu *et al.*, 2016] Zhiting Hu, Xuezhe Ma, Zhengzhong Liu, Eduard Hovy, and Eric Xing. Harnessing deep neural networks with logic rules. In *ACL*, pages 2410–2420, Berlin, Germany, 2016. Association for Computational Linguistics.
- [Jawahar *et al.*, 2019] Ganesh Jawahar, Benoît Sagot, and Djamé Seddah. What does bert learn about the structure of language? In *ACL*, pages 3651–3657, 2019.
- [Jimenez *et al.*, 2012] Sergio Jimenez, Claudia Becerra, and Alexander Gelbukh. Soft cardinality: A parameterized similarity function for text comparison. In *STARSEM and SemEval*, pages 449–453. Association for Computational Linguistics, 2012.
- [Kazmi and Schüller, 2016] Mishal Kazmi and Peter Schüller. Inspire at semeval-2016 task 2: Interpretable semantic textual similarity alignment based on answer set programming. In *SemEval*, pages 1109–1115, 2016.
- [Koehn *et al.*, 2003] Philipp Koehn, Franz Josef Och, and Daniel Marcu. Statistical phrase-based translation. In *NAACL-HLT*, pages 48–54. Association for Computational Linguistics, 2003.
- [Konopik *et al.*, 2016] Miloslav Konopik, Ondrej Prazak, David Steinberger, and Tomáš Brychcín. Uwb at semeval-2016 task 2: Interpretable semantic textual similarity with distributional semantics for chunks. In *SemEval*, pages 803–808, 2016.
- [Li and Srikumar, 2016] Tao Li and Vivek Srikumar. Exploiting sentence similarities for better alignments. In *EMNLP*, pages 2193–2203, 2016.
- [Li and Srikumar, 2019] Tao Li and Vivek Srikumar. Augmenting neural networks with first-order logic. In *ACL*, pages 292–302, Florence, Italy, July 2019. Association for Computational Linguistics.
- [Pagliardini *et al.*, 2018] Matteo Pagliardini, Prakhar Gupta, and Martin Jaggi. Unsupervised learning of sentence embeddings using compositional n-gram features. In *NAACL-HLT*, pages 528–540, 2018.
- [Pavlick *et al.*, 2015] Ellie Pavlick, Pushpendre Rastogi, Juri Ganitkevitch, Benjamin Van Durme, and Chris Callison-Burch. Ppdb 2.0: Better paraphrase ranking, fine-grained entailment relations, word embeddings, and style classification. In *ACL-IJCNLP*, pages 425–430, 2015.
- [Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher Manning. Glove: Global vectors for word representation. In *EMNLP*, pages 1532–1543, 2014.
- [Reimers and Gurevych, 2019] Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *EMNLP-IJCNLP*, pages 3973–3983, 2019.
- [Shao, 2017] Yang Shao. Hcti at semeval-2017 task 1: Use convolutional neural network to evaluate semantic textual similarity. In *SemEval*, pages 130–133, 2017.
- [Socher *et al.*, 2011] Richard Socher, Eric H Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y Ng. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *NeurIPS*, pages 801–809, 2011.
- [Speer *et al.*, 2017] Robert Speer, Joshua Chin, and Catherine Havasi. Conceptnet 5.5: An open multilingual graph of general knowledge. In *AAAI*, 2017.
- [Sultan *et al.*, 2015] Md Arafat Sultan, Steven Bethard, and Tamara Sumner. Dls@ cu: Sentence similarity from word alignment and semantic vector composition. In *SemEval*, pages 148–153, 2015.
- [Tai *et al.*, 2015] Kai Sheng Tai, Richard Socher, and Christopher D Manning. Improved semantic representations from tree-structured long short-term memory networks. In *ACL-IJCNLP*, pages 1556–1566, Beijing, China, 2015. Association for Computational Linguistics.
- [Towell *et al.*, 1990] Geoffrey G Towell, Jude W Shavlik, and Michiel O Noordewier. Refinement of approximate domain theories by knowledge-based neural networks. In *AAAI*, volume 861866. Boston, MA, 1990.
- [Vinyals *et al.*, 2015] Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. Pointer networks. In *NeurIPS*, pages 2692–2700, 2015.