SWIM: A Simple Word Interaction Model for Implicit Discourse Relation Recognition*

Wenqiang Lei[1], Xuancong Wang[2], Meichun Liu[3], Ilija Ilievski[1], Xiangnan He[1], Min-Yen Kan[1]
[1]National University of Singapore
[2]Institute for Infocomm Research
[3]City University of Hong Kong
{wenqiang, xiangnan, knmnyn}@comp.nus.edu.sg, wangxc@i2r.a-star.edu.sg
ilija.ilievski@u.nus.edu, meichliu@cityu.edu.hk

Abstract

Capturing the semantic interaction of pairs of words across arguments and proper argument representation are both crucial issues in implicit discourse relation recognition. The current state-of-the-art represents arguments as distributional vectors that are computed via bi-directional Long Short-Term Memory networks (BiLSTMs), known to have significant model complexity.

In contrast, we demonstrate that word-weighted averaging can encode argument representation which can be incorporated with word pair information efficiently. By saving an order of magnitude in parameters and eschewing the recurrent structure, our proposed model achieves equivalent performance, but trains seven times faster.

1 Introduction

Sentences alone do not serve to form coherent discourse. Logical relations, both inter- and intra-sententially, are needed for a coherent text. Such relations are termed discourse relations. Automatically recognizing discourse relations is useful for downstream applications such as machine translation and summarization.

Discourse relations can be overtly signaled by occurrences of *explicit* discourse connectives such as *Indeed* and *After that* (*cf* Ex. (3) & (4)). In contrast when the context is clear, such overt signals can be omitted, leading to discourse relations that it is said to be *implicitly* signaled (not marked by a lexical connective in the text; *cf* Ex. (1) & (2)). The lack of any overt signal makes implicit discourse relations much more challenging to recognize.

This explicit/implicit distinction is adopted by the Penn Discourse Treebank (PDTB, version 2.0) [Prasad *et al.*, 2008]. We adopted the PDTB for this study due to its large size when compared against other discourse corpora such as the Rhetorical Structure Theory Treebank (RST) [Carlson *et al.*, 2002]. While the PDTB has a hierarchical annotation scheme, currently most studies (e.g. [Chandrasekaran *et al.*, 2017]) – inclusive of this one – restrict their use to the topmost, Level–1 categories: *Contingency* (Ex. (1)), *Comparison* (Ex. (2)), *Expansion* (Ex. (3)) and *Temporal* (Ex. (4)). The two text spans where the discourse relation holds are called *arguments*, named Arg1 and Arg2 respectively. Arg2 is defined as the argument that syntactically houses the (explicit or implicit) discourse connective.

Ex (1) *You are so fortunate*. **The hurricane came five hours after you left**.

Ex (2) *In 1986, the number of cats was down to 1000.* **In 2002, it went up to 2000**.

Ex (3) *I never gamble too far*. In other words, **I quit after one try.**

Ex (4) *I was sleeping* when **he entered**.

Figure 1: Toy examples of each of the four Level–1 discourse relations annotated in the PDTB formalism. (1) and (2) are implicit relations; (3) and (4) are explicit. Arg1 is *italicized* and Arg2 is **bolded**, as per convention.

Modeling word pairs have been shown useful for implicit discourse relation recognition in many studies [Marcu and Echihabi, 2002a; Rutherford and Xue, 2014; Chen *et al.*, 2016]. This is because semantic interactions exist between the two arguments. This can be realised in many forms, of which word pairs are arguably the simplest. For example, the interaction between *up* and *down* is most likely to signal a *Comparison* relation as in Ex. (2). Traditional methods use word pairs [Marcu and Echihabi, 2002a], or variants like Brown Clustering pairs [Rutherford and Xue, 2014], as features for supervised learning.

In addition, *argument representation* — or how arguments are modeled as a whole — is also crucial for correct interpretation. Taking the *fortunate–hurricane* pair in Ex. (1), one might construe a *Comparison* relation due to the word pair's contrasting sentiment polarity. However, it is understood as a *Contingency* relation when the entire context of both arguments are taken into account.

To address the argument representation challenge, recent works have leveraged the powerful semantic representability of neural network models. For example, Gated Relevance

*Wenqiang Lei and Xuancong Wang are the contact authors.

Networks (GRN) [Chen *et al.*, 2016] and Neural Networks with Multi-Level Attention (NNMA) [Liu and Li, 2016], apply a bi-directional Long Short-Term Memory network (BiLSTM) [Hochreiter and Schmidhuber, 1997; Schuster and Paliwal, 1997] to represent each argument. Both models achieve the-state-of-the-art F_1 score without employing handcrafted features. However, BiLSTMs inevitably introduce many parameters. The large number of parameters slows the training process and is prone to overfitting. Especially in the context of the PDTB, a relatively small dataset, a lightweight representation is a viable method to simplify the model.

Motivated by this observation, we propose a new model that integrates the modeling of both word pair interaction and argument representation, without the use of BiLSTMs. Our model — termed the Simple Word Interaction Model (SWIM) — achieves a comparable F_1 score to the state-of-the-art and runs seven times faster by eschewing the use of BiLSTMs.

2 Related Work

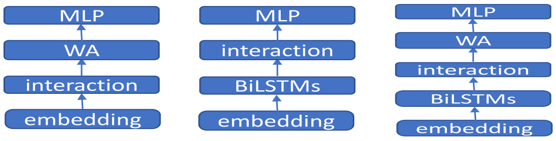
Supervised learning approaches for the implicit discourse relations recognition is the common paradigm in prior work. **Surface features** for the task include word pairs [Marcu and Echihabi, 2002b] sentiment polarity scores, General Inquirer tags [Pitler *et al.*, 2009], and parser production rules [Lin *et al.*, 2009]. Among all these models, the naïve feature of word pairs [Marcu and Echihabi, 2002a; Rutherford and Xue, 2014] – which is the co-occurrence frequency of a pair of words, one drawn from each of the two arguments – has proven to be extremely efficient.

Another trend is to employ **semi-supervised methods**. This is due to the limited amount of annotated data and the relative abundance of weakly-labeled data (*i.e.*, explicitly-signaled instances). For example, Lan et al. [2013] explored multitask learning; Hernault et al. [2010] applied feature vector extension; and Rutherford and Xue [2015] selectively added some explicit instances as implicit training data, according to their connectives.

The recent wave of **deep learning** approaches to NLP problems leverage word embeddings pre-trained on large corpora to achieve significant gains on tasks involving complex semantics. While many architectural designs have been explored (e.g. [Zhang *et al.*, 2016]), the state-of-the-art deep learning methods on our task — the GRN (discussed in detail later) and NNMA — are fundamentally BiLSTM based, involving many parameters, which leads to inefficiency in training and testing.

3 Simple Word Interaction Model (SWIM)

SWIM models both the fine-grained word pair interaction and coarse-grained argument representation. We first introduce the model here and detail its implementation in the following sections. To capture word pair interaction, we calculate an interaction score for each word pair that measures the importance of the interaction between its component words. For argument representation, we apply a weighted average of the component word pair representations. The argument representation thus encapsulates word pair interaction, and is passed to a final a multilayer perceptron layer (MLP) to determine the final discourse relation (*cf* Figure 2(a)).



(a) SWIM (b) GRN (c) BiLSTM + SWIM

Figure 2: The components of the SWIM, GRN and BiLSTM + GRN models. “WA” denotes “weighted average”.

3.1 Word Interaction Score

Following current best practices, SWIM's word interaction score captures both linear and quadratic relations between the two word's embeddings. Formally, let M and N denote the lengths of Argument 1 (hereafter, Arg1) and Argument 2 (Arg2). Further, let $\mathbf{x}_i$, $\mathbf{y}_j$ denote the pre-trained (row) word embeddings of i^{th} word of Arg1 and j^{th} word of Arg2, separately. Then for each pair of words $\mathbf{x}_i$ and $\mathbf{y}_j$, SWIM calculates an interaction score as in Eq. (1):

$$s_{ij} = \mathbf{x}_i \mathbf{A} {\mathbf{y}_j}^T + \mathbf{B}[\mathbf{x}_i, \mathbf{y}_j] + c_{ij} \qquad (1)$$

where $\mathbf{A} \in \mathbb{R}^{d \times d}$, $\mathbf{B} \in \mathbb{R}^{1 \times 2d}$ and $c_{ij} \in \mathbb{R}$ are trainable parameters, and $[\mathbf{x}_i, \mathbf{y}_j]$ is a concatenation of the two word embeddings. $\mathbf{x}_i \mathbf{A} {\mathbf{y}_j}^T$ models the quadratic relation between the two word embeddings, while $\mathbf{B}[\mathbf{x}_i, \mathbf{y}_j]$ captures linear relationships. c_{ij} is the bias term for the final interaction. At training, these parameters — $\mathbf{A}$, $\mathbf{B}$ and c_{ij} — model and encode word pair semantics, assigning a high interaction score when well correlated with a particular discourse relation class. Similar approaches have been applied to many fields, e.g. recommendation system [He *et al.*, 2017].

SWIM calculates the interaction matrix S, computing s_{ij} for each possible (i^{th}) word of Arg1 and (j^{th}) word of Arg2.

3.2 Argument Representation

Both (Bi)LSTMs and embedding averaging are valid methods for representing text sequences, inclusive of discourse arguments and sentences. Recent studies suggest that LSTMs perform well when sequential order is important [Iyyer *et al.*, 2015]. However, this is less the case in argument representation, where content plays a larger role than ordering – e.g., in Exs. (1) & (2), the discourse relations do not change even when the words are reordered. Simpler methods, such word averaging may be sufficient and effective as suggested by Wieting *et al.* [2016] who concluded that “word averaging models perform well for [the related tasks of] sentence similarity and entailment, outperforming LSTMs.” Hence, SWIM adopts embedding averaging.

However, embedding averaging alone is insufficient – each argument's words are actually understood in the context of the opposing argument. For example, on its own *down* in Arg1 of Ex. (2) is less likely considered to signal any discourse relation. However, once combined with the word *up* in Arg2, it becomes the most important signal in Arg1 for the *Comparison* relation. In light of this, SWIM represents each argument as an average of its component words' word embeddings, weighted for its interaction with the opposing argument.

We denote SWIM's argument representation for $Arg1$ ($Arg2$) as $\mathbf{x}'$ ($\mathbf{y}'$), as calculated in Eq. (2).

$$\begin{aligned}\mathbf{x}' &= \tfrac{1}{M}\textstyle\sum_{i=1}^{M}(\sum_{j=1}^{N}\frac{exp(s_{ij})}{\sum_{k=1}^{N}exp(s_{ik}))}[\mathbf{x_i},\mathbf{y_j}])\\ \mathbf{y}' &= \tfrac{1}{N}\textstyle\sum_{j=1}^{N}(\sum_{i=1}^{M}\frac{exp(s_{ij})}{\sum_{k=1}^{M}exp(s_{kj}))}[\mathbf{y_j},\mathbf{x_i}])\end{aligned} \tag{2}$$

In Eq. (2), we start with the concatenations of word pair embeddings $[\mathbf{x_i},\mathbf{y_j}]$, weighting them with the interaction score s_{ij}, to account for its importance.

Let us walk through Eq. (2), taking the computation of $\mathbf{x}'$ as an example. For each word $\mathbf{x_i}$ in Arg1, we enumerate all words in Arg2 ($\mathbf{y_j}$ for $j \in \{1, 2...N\}$) to form word pairs ($[\mathbf{x_i},\mathbf{y_j}]$). We weight these word pair representation according to its normalized interaction score, obtaining an interaction weighted word representation, the term in the parentheses. All M interaction-enhanced word representations are averaged to arrive at the final form for $\mathbf{x}'$.

SWIM computes a single representation of both Arg1 and Arg2 by concatenating $\mathbf{x}'$ and $\mathbf{y}'$ for input to the final MLP classification to obtain the $\mathbf{output}$ discourse relation:

$$\mathbf{output} = f_o(\mathbf{W}_o f_h(\mathbf{W}_h[\mathbf{x}',\mathbf{y}'])) \tag{3}$$

where $\mathbf{W}_h \in \mathbb{R}^{4d\times k}$ and $\mathbf{W}_o \in \mathbb{R}^{k\times n}$. In $\mathbf{W}_o$ and $\mathbf{W}_h$, n is the number of class labels, d is the embedding size, k is the size of the hidden layer. f_h and f_o are sigmoid activation functions. The final classification layer described in Eq. (3) is a two-layer MLP whose hidden layer is designed to obtain a more abstract representation.

3.3 Model Discussion On BiLSTMs

Both state-of-the-art prior models for implicit discourse relation classification – GRN and NNMA – adopt BiLSTMs for argument representation. A discussion on the role of BiLSTMs in both models is relevant. We use GRN as a sample for this discussion as i) both GRN and our proposed SWIM is designed to model word pair interaction and apply similar word interaction calculation approach; ii) the necessity of BiLSTMs in GRN has been previously studied in [Chen *et al.*, 2016].

The GRN workflow is illustrated in Figure 2(b). GRN feeds one argument into an individual BiLSTMs to get an intermediate representation for each word, denoted as $\mathbf{h}_{x_i}$ and $\mathbf{h}_{y_j}$ for each $\mathbf{x}_i$ and $\mathbf{y}_j$, separately. As illustrated in Figure 3, the intermediate representation incorporates information from the argument as a whole.

For each pair of $\mathbf{h}_{x_i}$ and $\mathbf{h}_{y_j}$, an interaction score s_{ij} is calculated as in Eqs. (4) & (5). As we focus on the role of BiLSTMs, we do not discuss interaction scores further.

$$\begin{aligned}s_{ij} = \mathbf{u}(g \odot \mathbf{h}_{x_i}\mathbf{M}^{[1:r]}\mathbf{h}_{y_j}^{T} +\\ (1-g)\odot f(\mathbf{V}[\mathbf{h}_{x_i},\mathbf{h}_{y_i}]+\mathbf{b})\end{aligned} \tag{4}$$

$$g = \sigma(\mathbf{W}_g[\mathbf{h}_{x_i},\mathbf{h}_{y_i}]) \tag{5}$$

In the above, $\odot$ denotes element-wise multiplication. $\mathbf{M}^{[1:r]} \in \mathbb{R}^{r\times 2d\times 2d}$, $\mathbf{V} \in \mathbb{R}^{r\times 4d}$, $\mathbf{W}_g \in \mathbb{R}^{r\times 4d}$, $\mathbf{u}, \mathbf{b} \in \mathbb{R}^{r}$ are trainable parameters where d is the embedding size and $r = 2$ according to the original paper [Chen *et al.*, 2016]. Both f and σ are sigmoid activation functions. Eq. (4) & (5) of GRN degrades to Eq. (1) in SWIM if we set $r = 1$, $g = 0.5$ and remove aggregating vector $\mathbf{u}$. Similar to SWIM, a word pair interaction matrix $\mathbf{S} \in \mathbb{R}^{M\times N}$ is obtained. Finally, the matrix $\mathbf{S}$ is max pooled and then reshaped into a vector to feed a final multi-layer perceptron (MLP) for classification.

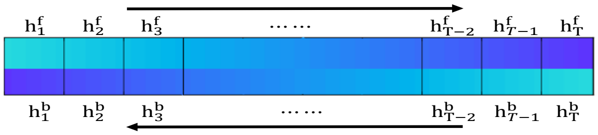

$\mathrm{h}_1^{\mathrm{f}}$
$\mathrm{h}_2^{\mathrm{f}}$
$\mathrm{h}_3^{\mathrm{f}}$
……
$\mathrm{h}_{\mathrm{T}-2}^{\mathrm{f}}$
$\mathrm{h}_{T-1}^{\mathrm{f}}$
$\mathrm{h}_{\mathrm{T}}^{\mathrm{f}}$
$\mathrm{h}_1^{\mathrm{b}}$
$\mathrm{h}_2^{\mathrm{b}}$
$\mathrm{h}_3^{\mathrm{b}}$
……
$\mathrm{h}_{\mathrm{T}-2}^{\mathrm{b}}$
$\mathrm{h}_{T-1}^{\mathrm{b}}$
$\mathrm{h}_{\mathrm{T}}^{\mathrm{b}}$

Figure 3: An illustration of BiLSTM for argument representation. A BiLSTM treats one argument as a sequence of words $\mathbf{x}_1, \mathbf{x}_2...\mathbf{x}_T$. It consists of two LSTMs, one for forward propagation (from $\mathbf{x}_1$ to $\mathbf{x}_T$), outputting $\mathbf{h}_1^f...\mathbf{h}_T^f$ and one for backward propagation (from $\mathbf{x}_T$ to $\mathbf{x}_1$), outputting $\mathbf{h}^b{}_T...\mathbf{h}^b{}_1$. The information continue to accumulate as it propagates. The darker shade of color signifies that more information has been accumulated in the current sequence during propagation. For each word $\mathbf{x}_i$, the final representation is the concatenation of both direction's output $\mathbf{h}_i = [\mathbf{h}^f{}_i, \mathbf{h}^b{}_i]$. Therefore, $\mathbf{h}_i$ contains the information of the whole argument.

The matrix $\mathbf{S}$, which is the representation for each instance, actually contains two aspects of information: i.) word content information (whether significant word pair interactions exist); and ii) positional information (where the strong interacting words are). Thus, BiLSTMs in GRN contribute both position information and content information. However, we can eschew BiLSTM use by only handling content information via embedding average, since embedding average and BiLSTM are know to have similar content representations. We can test this hypothesis by inserting BiLSTMs into SWIM to evaluate its effect: we plug a BiLSTM layer in between the embedding layer and word interaction layer, as illustrated in Figure 2. We refer to this model as the *BiLSTM + SWIM* model in the following discussion[1].

To evaluate BiLSTMs' effect on model complexity, we further calculate the number of parameters of various model. As shown in Tabel 1, the use of BiLSTMs, which encode superfluous position information, increase the model complexity by an order of magnitude (Row 1, 2, 5, 6). As for NNMA, it contains a complex multi-level attention layer unlike BiLSTMs which is less different from our design. Therefore, we just list its number of parameters for reference without discussing it in more details.

4 Experiments

We now evaluate SWIM, with the aim to show its competitiveness in terms of prediction performance. Importantly, we note that the evaluation of a computational model not only lies in the best prediction performance, but also its stability (robustness), and efficiency. This is especially important for neural network models, which are computationally intensive

[1] We use a dense layer between BiLSTM and SWIM to reduce the output size from $2d$ to d, to lessen the number of parameters in BiLSTM + SWIM.

Model	$d = 50$	$d = 100$
1. BiLSTM	80K	320k
2. GRN	100k	400k
3. NNMA (2 levels)	260k	720k
4. NNMA (3 levels)	350k	920k
5. SWIM	**12.5k**	**30k**
6. BiLSTM + SWIM	102.5k	390k

Table 1: Model complexity analysis. Number of parameters versus the hidden size d. We omit matrices with fewer than $1k$ parameters.

and sensitive to random factors. We first describe the systems we compare against, then describe the experimental setup before describing the main results.

4.1 Comparison Systems

In the following list of architectures, numberings follow those used in Table 2 for convenience. First, we must validate performance against the current state-of-the-art baselines in terms of raw prediction performance, in the numbered systems below. The cited papers, our SWIM and replicated experiments follows the same training, development, and testing splits, so are comparable.

1-2. Gated Relevance Networks (GRN; [Chen *et al.*, 2016]). We replicate their architecture with the assistance of the original author, tuning parameters according to the original paper.

3-4. Neural Networks with Multi-Level Attention (NNMA; [Liu and Li, 2016]). We cite results from the paper.

We study a number of variants of SWIM to test our design decisions. In the lettered systems below, we examine variations on argument representation, using embedding averaging and BiLSTMs:

a. Embedding Average (EA). We perform naïve embedding averaging to get a representation vector for each argument. Then, we concatenate the two vectors for input to a MLP as in Eq. (3).

b. BiLSTM. We use the word embeddings as input to BiLSTMs (h_T^f, h_1^b), concatenate their output to form the argument representation and pass it to the MLP layer for classification[2]. This setting benchmarks the vanilla use of BiLSTM for argument representation.

c. Word Interaction Score (WIS). We use scores in the interaction matrix $\mathbf{S}$ for classification without modification. Following the GRN approach (*cf* Section 3.3), we perform max-pooling on $\mathbf{S}$ before inputting the scores to the MLP layer. This tests whether our weighted argument representation helps.

d. BiLSTMs + WIS. Following GRN, we feed word embeddings to BiLSTMs to get an intermediate representation. Then we calculate word interaction scores based on these intermediate representation using Eq. (1), obtaining the interaction matrix $\mathbf{S}$ which is max-pooled and fed into a MLP layer. This aims to test the effect of BiLSTMs on WIS. BiLSTM + WIS is similar to GRN except for the word interaction computation part. However, we here only focus on the study of BiLSTMs without discussion its difference.

e. BiLSTM + SWIM. As described in Section 3.3, this tests whether BiLSTMs further aid SWIM, since content information has already been encoded by SWIM's word embedding averaging.

We also investigate whether both linear and quadratic terms are necessary for SWIM's word interaction computation:

A. Quadratic Term (QT only). Only the quadratic term and bias term of Eq. (1) are retained.

B. Linear Term (LT only). Only the linear and bias terms are retained.

4.2 Experimental Setting

We adopt the standard settings for the PDTB v2.0 dataset use in our experimentation (Sections 2–20, Sections 0-1 & Sections 23-24, and Sections 21-22 for training, development and testing, respectively.

We follow the practice of [Zhou *et al.*, 2010; Liu and Li, 2016] which follow the standard definition of PDTB v2.0, which admits a separate categorization outside of discourse relations for *EntRel* (hence distinct from *Expansion*), unlike [Pitler *et al.*, 2009; Chen *et al.*, 2016] that deemed entity relations (*EntRel*) as a form of *Expansion* relations. Aside from this, our other settings are standard: we use Stanford CoreNLP [Manning *et al.*, 2014] for tokenization, pad all sentences to length 50, and use Stanford's GloVe [Pennington *et al.*, 2014] 100 dimensional pre-trained word embeddings for SWIM and 50 dimensional pre-trained embedding for BiLSTMs + SWIM. The embedding layer is fixed during training, and dropout is performed on the input and MLP layers (dropout percentage = 20%). For training, we adopt multi-class cross-entropy loss, using AdaGrad for the stochastic optimization [Duchi *et al.*, 2011]. The initial learning rate is set at 0.01, with a batch size of 32. Following [Liu and Li, 2016; Rutherford and Xue, 2014], we use instance re-weighting.

4.3 Experimental Results

Table 2 presents our results. Prediction performance is evaluated in both standard schemes of n binary classifications and a single n-way classification (where $n = 4$), using macro F_1.

The upper portion of Table 2 shows that our SWIM architecture (Row 5) performs on par with GRN and NNMA, with the exception of *Temporal* relations. Introspecting these results at an instance level, we find that GRN, NNMA and SWIM exhibit little agreement on *Temporal* relations. As *Temporal* relations only constitute 5% of the dataset, it is a minority class where its performance may be largely affected by random factors. In our efficiency analysis, we calculate the time cost per instance in one training epoch (measured in milliseconds). Our proposed SWIM runs 7 times (3.46/0.49) faster than GRN.

[2] Another common approach for BiLSTM argument representation is to get the intermediate representation for every time step $\mathbf{h}_i, i \in \{1...T\}$ and average these intermediate representation as EA does. We find that both approaches yield similar results, hence only the approach in the body text is reported.

Model	Comp.	Cnt.	Exp.	Temp.	4-way	Time (ms)	Avg.Std.
1. GRN (cited)	40.17%	54.76%	–	31.32%	–	–	–
2. GRN (replicated)	39.05%	54.53%	69.01%	33.52%	44.61%	3.46	0.012
3. NNMA (cited, two level)	39.86%	53.69%	69.71%	37.61%	44.95%	–	–
4. NNMA (cited, three level)	36.70%	54.48%	**70.43%**	**38.84%**	46.25%	–	–
5. SWIM	**40.47%**	**55.36%**	69.50%	35.34%	**46.46%**	0.49	0.008
a. EA	33.01%	46.56%	68.31%	29.59%	40.22%	0.08	0.005
b. BiLSTM	34.01%	47.31%	68.53%	30.01%	40.84%	3.00	0.011
c. WIS	35.53%	51.15%	68.98%	30.72%	40.64%	0.45	0.007
d. BiLSTM + WIS	39.42%	53.85%	70.05%	32.75%	44.57%	3.37	0.011
e. BiLSTM + SWIM	38.71%	54.32%	70.02%	35.06%	45.96%	3.37	0.012
A. QT only	37.73%	53.02%	68.12%	33.21%	43.25%	0.34	0.008
B. LT only	36.78%	51.51%	67.41%	32.72%	42.04%	0.18	0.008

Table 2: Models' effectiveness, efficiency and stability. Effectiveness measured by macro F_1; efficiency, by average processing time per instance, in milliseconds; stability, by macro-averaged standard deviation over 10 runs.

This result is significant as the training times listed in Table 2 are for individual epochs and for a single architecture. The time savings benefits two aspects: first, neural models are typically trained over hundreds of epochs and with hundreds of individual configurations for hyperparameter tuning, leading to significant savings in development and tuning times in creating the final model. Second, semi-supervised learning is an important direction for implicit discourse relation recognition, due to the abundance of weakly labeled data. In semi-supervised learning, usually a much large number of training instances are involved. For example, Liu *et al.* [2016] report training with over 400k instances. With 400k instances, we estimate that a GRN architecture will take about 38 hours for a run of $\sim$100 epochs; in comparison, SWIM would complete in less than 6 hours.

Model stability is another concern. In our analysis we run each model 10 times by initializing with different random seeds for both 4 binary classifications and the single, four-way classification, calculating the standard deviation over the ten runs. We see that SWIM's predictive performance is more stable (last column, comparing 0.008 with 0.012, F-test $p = 0.001984 < 0.005$) than GRN.

The middle portion of the table provides empirical validation of SWIM's choice of argument and word pair representations. Rows a and b validate our hypothesis that embedding averaging achieves prediction performance comparable to the more complex BiLSTM model. Simplifying SWIM's model further by employing only the raw word interaction score (Row c) also underperforms.

Could the simplified treatment of the interaction scores coupled with the standard BiLSTM approach close the performance gap? Row d explores this option. The answer is 'no', as it underperforms; SWIM (Row 5) adds the use of embedding averaging to sensitize the argument representation to the interaction scores. Augmenting SWIM with BiLSTMs does not improve the results further (Row e), plausibly supporting our hypothesis that positional information encoded by BiLSTMs may contribute noise over the use of the content cues already encoded by embedding averaged argument representation. Additionally, all BiLSTMs models take significantly longer training times and produce less stable results (*cf* Row 5 vs. Rows 2, b, d, and e).

	GRN	SWIM
$test'$	42.35% (-2.26%)	46.46% (NC)

Table 3: Word order sensitivity comparison: F_1 of a single 4-way classification on the scrambled ($test'$) set.

Finally, comparing SWIM against the selective use of only the quadratic or linear word interaction terms in the bottom portion of the table suggest that both are needed to properly model word interaction (Row 5 vs. Rows A and B).

Overall, we conclude that the current SWIM configuration strikes a good balance among effectiveness, stability, efficiency and model complexity.

Discussion

There are loose ends to validate in our claims that have not yet been supported by evidence. Here, we detail three issues where we present auxiliary data to buttress our arguments.

First, we have argued that positional information encoded by BiLSTM (in the guise of GRNs) is not needed for our classification task. We validate this proposition via supplemental experimentation. To give credence that GRN models positional information, we randomize the order of the words in each argument in the test set to obtain a new test set ($test'$). We then test the pre-trained GRN, SWIM against $test'$. Table 3 gives the 4-way classification performance figures on $test'$. The results indicate that GRN is sensitive to the position of words, while SWIM-based models retain their performance, validating that SWIM does not model positional information.

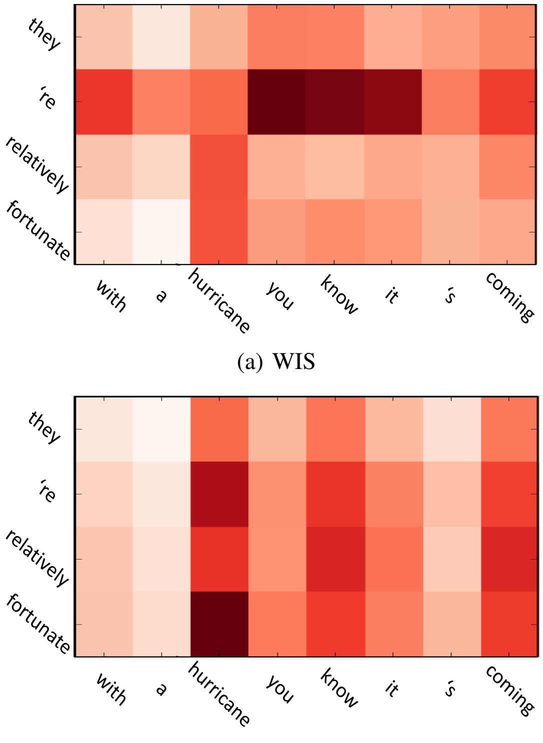
Second, we need to assess the sensitivity of the neural architectures to dataset scale, as we have argued that the SWIM architecture is optimal for the task, independent of the scale of data currently available. We train and test GRN and SWIM with different model sizes by changing the hidden layer and embedding sizes[3] of each model to have their total number of

[3] For GRN, $50d$ is the smallest pre-trained GloVe word embed-

Model	10k	30k	100k	300k
GRN	42.06%	43.03%	**44.61%**	43.59%
SWIM	45.93%	**46.46%**	44.32%	43.95%

Table 4: F_1 score of 4-way classification with different total number of parameters. The optimal configuration is reported in Table 2, and bolded per system.

parameters be at parity: set to 10k, 30k, 100k and 300k (Table 4). This result suggests that GRN intrinsically requires much more parameters to achieve better prediction results; in contrast, SWIM makes better use of its limited number of parameters.



(a) WIS

(b) SWIM

Figure 4: The interaction scores of one PDTB instance (`wsj_1691`), generated by both WIS (a) and SWIM (b). Darker entries indicate a higher interaction score.

Finally, we also wish to assess whether SWIM actually does assigns high interaction scores to important word pairs in the matrix **S** through per instance analysis. In Figure 4, we introspect the interaction matrices of WIS and SWIM on a PDTB example to illustrate how they differently they behave. We note that both WIS and SWIM calculate word interaction scores in the same way, but that SWIM's contextualization of its argument representation modeling assigns high scores more intuitively than WIS does. WIS assigns word pairs featuring *are ('re)* high scores, possibly due to its corpus frequency. The arithmetic post-processing of the argument context by SWIM redistributes scores, correctly assigning the *fortunate–hurricane* pair the highest interaction score. Separately, Figure 5 shows SWIM's interaction scores on a *Comparison* instance, where the words (row) *down* and (column) *up* are assigned high weights, which is in accord with our intuition that the word pair is critical in determining the discourse relation. Interestingly, the word (row) *about* also yields a high score. What was the cause for this? Drilling down, we find that the *Wall Street Journal* uses similar textual expressions "[numbers] ... increase/decrease/up/down ... about" when reporting many instances of financial news. The high weight for *about* lends evidence that SWIM can discover such important contextual co-occurrence corpus patterns.

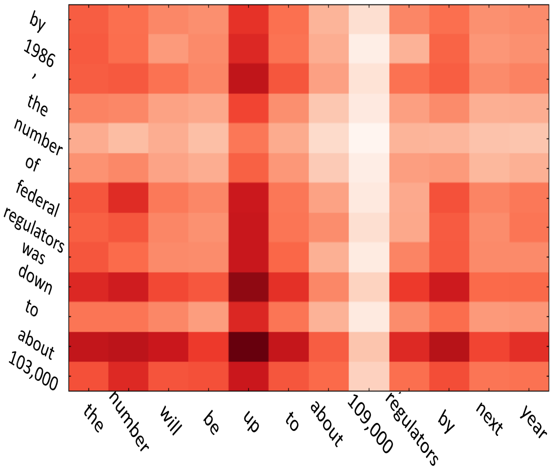


Figure 5: The interaction scores of a *Comparison* PTDB instance (`wsj_1499`) generated by SWIM.

5 Conclusion

We proposed a simple neural model for implicit discourse relation recognition, named SWIM, which accounts for both word pair interaction and argument representation. In contrast to previous works, we utilize word embedding average, instead of BiLSTMs, for argument representation. Experiment results show our model is more stable and runs faster while still achieving state-of-the-art F_1 scores.

In the wider context of neural network research, our work finds additional evidence that BiLSTMs spend much effort to model positional information, which we have shown to be less helpful for our task. In tasks where word content is valued over word ordering information, our work suggests that simpler models such as an embedding averaging, can replace BiLSTMs while achieving similar performance.

Acknowledgments

We gratefully acknowledge Qian Zhong of the Department of Linguistics and Translation, City University of Hong Kong for fruitful discussions on this research, and also acknowledge the support of NVIDIA Corporation for their donation of the Titan X GPU that facilitated this research.

ding. Therefore, we have to reduce the hidden size of GRN. For SWIM, we use larger pre-trained embeddings and enlarge the hidden size to increase model size.

References

[Carlson *et al.*, 2002] Lynn Carlson, Mary Ellen Okurowski, and Daniel Marcu. *RST discourse treebank*. Linguistic Data Consortium, University of Pennsylvania, 2002.

[Chandrasekaran *et al.*, 2017] Muthu Kumar Chandrasekaran, Carrie Demmans Epp, Min-Yen Kan, and Diane Litman. Using discourse signals for robust instructor intervention prediction. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, pages 3415–3421, 2017.

[Chen *et al.*, 2016] Jifan Chen, Qi Zhang, Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. Implicit discourse relation detection via a deep architecture with gated relevance network. *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, 2016.

[Duchi *et al.*, 2011] John Duchi, Elad Hazan, and Yoram Singer. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159, 2011.

[He *et al.*, 2017] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *26th International World Wide Web Conference*, 2017.

[Hernault *et al.*, 2010] Hugo Hernault, Danushka Bollegala, and Mitsuru Ishizuka. A semi-supervised approach to improve classification of infrequent discourse relations using feature vector extension. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 399–409. Association for Computational Linguistics, 2010.

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

[Iyyer *et al.*, 2015] Mohit Iyyer, Varun Manjunatha, Jordan Boyd-Graber, and Hal Daumé III. Deep unordered composition rivals syntactic methods for text classification. In *Proceedings of the Association for Computational Linguistics*, 2015.

[Lan *et al.*, 2013] Man Lan, Yu Xu, Zheng-Yu Niu, et al. Leveraging synthetic discourse data via multi-task learning for implicit discourse relation recognition. In *Proceedings of the 51th Annual Meeting on Association for Computational Linguistics*, pages 476–485. Citeseer, 2013.

[Lin *et al.*, 2009] Ziheng Lin, Min-Yen Kan, and Hwee Tou Ng. Recognizing implicit discourse relations in the penn discourse treebank. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*, pages 343–351. Association for Computational Linguistics, 2009.

[Liu and Li, 2016] Yang Liu and Sujian Li. Recognizing implicit discourse relations via repeated reading: Neural networks with multi-level attention. *EMNLP*, 2016.

[Liu *et al.*, 2016] Yang Liu, Sujian Li, Xiaodong Zhang, and Zhifang Sui. Implicit discourse relation classification via multi-task neural networks. *AAAI*, 2016.

[Manning *et al.*, 2014] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *ACL (System Demonstrations)*, pages 55–60, 2014.

[Marcu and Echihabi, 2002a] Daniel Marcu and Abdessamad Echihabi. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 368–375. Association for Computational Linguistics, 2002.

[Marcu and Echihabi, 2002b] Daniel Marcu and Abdessamad Echihabi. An unsupervised approach to recognizing discourse relations. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 368–375. Association for Computational Linguistics, 2002.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove: Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–43, 2014.

[Pitler *et al.*, 2009] Emily Pitler, Annie Louis, and Ani Nenkova. Automatic sense prediction for implicit discourse relations in text. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 683–691. Association for Computational Linguistics, 2009.

[Prasad *et al.*, 2008] Rashmi Prasad, Nikhil Dinesh, Alan Lee, Eleni Miltsakaki, Livio Robaldo, Aravind K Joshi, and Bonnie L Webber. The penn discourse treebank 2.0. In *LREC*. Citeseer, 2008.

[Rutherford and Xue, 2014] Attapol Rutherford and Nianwen Xue. Discovering implicit discourse relations through brown cluster pair representation and coreference patterns. In *EACL*, volume 645, page 2014, 2014.

[Rutherford and Xue, 2015] Attapol Rutherford and Nianwen Xue. Improving the inference of implicit discourse relations via classifying explicit discourse connectives. In *Proceedings of the NAACL-HLT*, 2015.

[Schuster and Paliwal, 1997] Mike Schuster and Kuldip K Paliwal. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*, 45(11):2673–2681, 1997.

[Wieting *et al.*, 2016] John Wieting, Mohit Bansal, Kevin Gimpel, and Karen Livescu. Towards universal paraphrastic sentence embeddings. *ICLR*, 2016.

[Zhang *et al.*, 2016] Biao Zhang, Deyi Xiong, and Jinsong Su. Variational neural discourse relation recognizer. *EMNLP*, 2016.

[Zhou *et al.*, 2010] Zhi-Min Zhou, Yu Xu, Zheng-Yu Niu, Man Lan, Jian Su, and Chew Lim Tan. Predicting discourse connectives for implicit discourse relation recognition. In *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*, pages 1507–1514. Association for Computational Linguistics, 2010.