Externally Controllable RNN for Implicit Discourse Relation Classification

Xihan Yue, Luoyi Fu, and Xinbing Wang

Department of Computer Science and Engineering
Shanghai Jiao Tong University, China
{yuexihan, yiluofu, xwang8}@sjtu.edu.cn

Abstract. Without discourse connectives, recognizing implicit discourse relations is a great challenge and a bottleneck for discourse parsing. The key factor lies in proper representing the two discourse arguments as well as modeling their interactions. This paper proposes two novel neural networks, i.e., externally controllable LSTM (ECLSTM) and attention-augmented GRU (AAGRU), which can be stacked to incorporate arguments' interactions into their representing process. The two networks are variants of Recurrent Neural Network (RNN) but equipped with externally controllable cells that their working processes can be dynamically regulated. ECLSTM is relatively conservative and easily comprehensible while AAGRU works better for small datasets. Multilevel RNN with smaller hidden state allows critical information to be gradually exploited, and thus enables our model to fit deeper structures with slightly increased complexity. Experiments on the Penn Discourse Treebank (PDTB) benchmark show that our method achieves significant performance gain over vanilla LSTM/CNN models and competitive with previous state-of-the-art models.

Keywords: implicit discourse relation classification, recurrent neural network, sequence pair modeling

1 Introduction

A text span may connect to another span when there is a causal relation between them or when they contrast each other. Such semantic relations are termed rhetorical or discourse relations [5]. Discourse parsing, the process of which is to understand the internal structure of a text and identify the discourse relations in between its segments, is a fundamental task in Natural Language Processing (NLP) since it benefits a lot of downstream applications such as information retrieval, question answering and automatic summarization [11].

Discourse connectives (e.g. *and*, *because*, etc) are considered as one of the most critical linguistic cues for discourse relations. Depending on whether there are connectives in between text arguments, discourse relations can be categorized into implicit and explicit ones. According to Pitler [3], an over 90% of accuracy rate can be achieved in classifying the explicit relations, so that the bottleneck of discourse parsing lies in recognizing implicit relations.

Conventional methods using one-hot representations [4, 7] and recent neural network (NN) models [13, 15] using dense real-value representations, despite their differences in feature representations, all follow a strategy that decomposes the process to two independent steps, modeling the two discourse arguments and then modeling their interactions. Intuitively, these methods simulate the single-pass reading process [19]. However, a large number of irrelevant components in texts make crucial information easily concealed. **Without specific learning aims and guidances, single-pass reading is heavily affected by data sparsity while still inadequate to capture comprehensive representations of the text arguments.**

In order to solve the dilemma, we leverage the intuition that critical information could be dynamically exploited through several passes of reading [19]. Specifically, we use previously obtained argument representations as guidance and reread the texts to gradually get deeper and preciser understandings. Now, let us check one real example to elaborate the new strategy.

[*Arg1*]: The World Psychiatric Association voted at an Athens parley to conditionally readmit the Soviet Union.

[*Arg2*]: Moscow could be suspended if the misuse of psychiatry against dissenters is discovered during a review within a year.

[*Implicit Connective*]: However

[*Discourse relation*]: Contrast

In the above example, each argument is composed of multiple phrases containing different meanings that without information from the other one we can only allocate identical attentions to them. In the second stage of reading, we have guidance and can retain only the most relevant parts, reaching the conclusion that there exists a relation rather than no relation. In order to discriminate different relation types, we further move to the third stage of reading where we term the process of gradually capturing the most relevant information for classification as mutually guided sequence pairs modeling. Here, we use "sequence pairs" rather than "argument pairs" to emphasize that the guidance mechanism acts upon the sequential processing of words and our method can be generally adopted on sequences-interaction related tasks.

We note that the key to implement the repeated reading strategy is finding a proper model for the process of guided regenerating. Since Recurrent Neural Network (RNN) is the most natural way to model sequence, we customize it by equipping its internal computational unit with externally controllable gates. Gated RNNs such as Long Short-Term Memory (LSTM) are not suitable for naive stacking. Usually, outputs of lower-level LSTM are weighted and summed and feed as input to higher-level LSTM to model the hierarchical structure of articles [17]. Here, we take several layers of RNNs with their inputs directly connected to the original text arguments and use lower layer's outputs as higher layer's guidance to integrate them into a multilevel structure. In this paper, we propose two novel recurrent neural networks to implement the strategy of mutu-

ally guided modeling. One is Externally Controllable Long Short Term Memory (ECLSTM), whose internal gates can be controlled by externally supplied vector. The other is Attention-Augmented Gated Recurrent Unit (AAGRU), which uses attention mechanism to augment traditional GRU. The difference between ECLSTM and AAGRU lies in that AAGRU renders supplied vector stronger controlling power and bears a simpler structure. In summary, this paper makes the following contributions:

1. We propose stacked RNN to implement the repeated reading strategy and prove its practicability in experiments.
2. We design ECLSTM as a conservative extension of LSTM and AAGRU as a strengthened version. Their efficiencies are both empirically verified in experiments and AAGRU exhibits better performance in small datasets.

2 Related Work

The first formal study of implicit discourse classification dates back to Marcu and Echihabi (2002) [1], which proposes a method for cheap acquisition of training data. However, their idea, which deletes connectives in unambiguous explicit patterns and treats the remaining spans as belonging to implicit relations, has been proven to be inadequate to generate realistic examples [4]. The interest in implicit discourse parsing surges since the release of PDTB [2], a resource of annotated discourse relations including implicit ones.

Conventional methods for implicit relations classification use hand-crafted linguistic features [4, 7]. Those methods, which make heavy use of word pairs, suffer from data sparsity problems. Recently, unsupervised dense real-value word representations are demonstrated to outperform previous one-hot representations [12, 21], and neural network models, which alleviate the need for traditional extensive feature engineering, can achieve even better performance [13, 15].

Incorporating arguments' interactions into their modeling processes is not a new concept since the use of word pairs naively implements the strategy. Chen [15] extends the naive methods by utilizing dense representations and contextual informationinformation, which augments word vectors with contextual information through a bidirectional LSTM neural network and generates pair-wise representations with those newly calculated vectors. Liu and Li [19] proposes neural networks, which consist of one bidirectional LSTM layer and multiple attention layers, to mimic the repeated reading strategy. Rather than simply adding attention layers, we merge attention mechanism in RNN units that enables our model to fit deeper structures with little increased complexity.

Another notable line aims at building multi-model approaches and using external resources to help improve accuracy. Multi-task neural network model, which share partial architecture for implicit discourse classification tasks of different corpora, is proposed to alleviate the shortage of labeled data [18, 16]. Qin [20] proposes an adversarial network, which leverages implicit connectives manually added in PDTB, as an regularization mechanism to adaptively regularize

parameters. Those approaches can be seen as integrated frameworks, and our methods can be used as replacement parts.

3 Methods

The strategy proposed in this paper is to incorporate arguments' interactions into their modeling process to dynamically exploit critical information. Firstly, each arguments are independently processed to get a general sense, then we reprocess the argument pairs under the guidance of the general understanding to get more relevant representations to the recognition task. The newly acquired representations are then used as guidance for further processing. In practice, we limit the upper bound of total guiding and reprocessing times to 2.

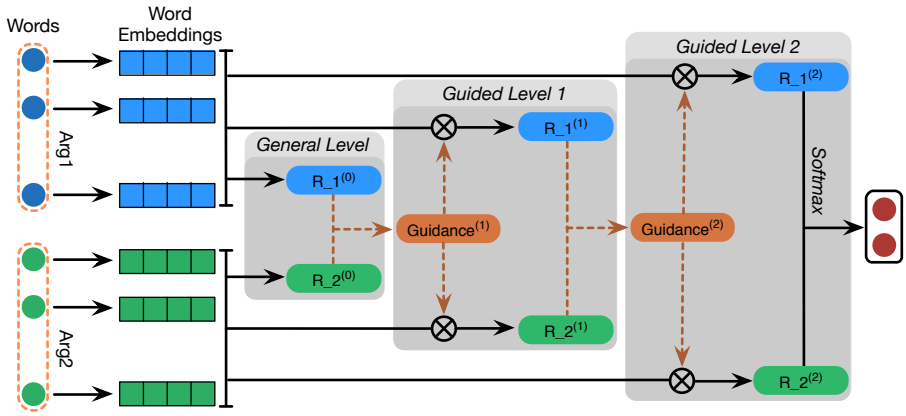


Fig. 1: Architecture of the Model.

3.1 Model Architecture

The overall architecture of our model is illustrated in Figure 1. Let us take $(Args, y)$ as a pair of input and output, where $Args = (Arg_1, Arg_2)$ is the argument pair and y is the golden standard relation. The two arguments contain different amounts of words:

$$Arg_1 = [w_{1,1}, w_{1,2}, \ldots, w_{1,L_1}], \quad Arg_2 = [w_{2,1}, w_{2,2}, \ldots, w_{2,L_2}]$$

Initially, we associate each lemma in lemma vocabulary with a vector representation $e_l \in R^{D_l}$ and each part of speech (POS) tag in POS vocabulary with a vector representation $e_p \in R^{D_p}$. Firstly, words in arguments are converted to word embeddings by concatenating their corresponding lemma vectors and pos vectors:

$$x_{i,j} = e_{L(w_{i,j})} \oplus e_{P(w_{i,j})}$$

where $x_{i,j} \in R^{D_e}$ is word embedding vector of $w_{i,j}$. L denotes the operation of lemmatization, P represents the operation of POS tagging and $D_e = D_l + D_p$ is the dimension of word embedding.

Let $E(Arg_i)$ denote the process of transforming Arg_i's words to their vector representations, we have:

$$X_1 = E(Arg_1) = [x_{1,1}, x_{1,2}, \ldots, x_{1,L_1}], \quad X_2 = E(Arg_2) = [x_{2,1}, x_{2,2}, \ldots, x_{2,L_2}]$$

Then we calculate the general-level representations of arguments as:

$$R_1^{(0)} = g^{(0)}(X_1), \quad R_2^{(0)} = g^{(0)}(X_2)$$

In implementing $g^{(0)}$, we separately adopt one-dimensional CNN and bi-LSTM with attention to compare their effects. Now we can obtain the level-1 guidance vector by concatenating $R_1^{(0)}$ and $R_2^{(0)}$:

$$Guidance^{(1)} = R_1^{(0)} \oplus R_2^{(0)}$$

Having obtained the guidance vector, we re-calculate the representations of the argument pairs:

$$R_1^{(1)} = g_1^{(1)}(X_1, Guidance^{(1)}), \quad R_2^{(1)} = g_2^{(1)}(X_2, Guidance^{(1)})$$

where $R_i^{(1)}$ is the level-1 representation for Arg_i. We adopt RNN with externally controllable cells as $g_i^{(1)}$ to enable guidance vector to control the sequential processing of words in Arg_i. After K-th repeating of the guided level, we use the newest representations derived from the top level to recognize the discourse relation through a fully connected softmax layer:

$$P = softmax(W_p(R_1^{(K)} \oplus R_2^{(K)}) + b_p)$$

3.2 General Sequence Pairs Modeling

The operations of general modeling the two arguments will be the same since it conforms to common sense and reduces parameters. So we treat X_1 and X_2 obtained from the embedding-lookup layer as a unified form $X = [x_1, x_2, \ldots, x_L]$.

Here we briefly introduce two methods, which are one-dimensional CNN with max pooling and bi-LSTM with attention.

One-Dimensional Convolutional Neural Network has been broadly used for modeling sequences. Filter matrices $[W_1, W_2, \ldots, W_k]$ with variable sizes $[l_1, l_2, \ldots, l_k]$ are utilized to perform convolutional operations. The argument embeddings will be transformed to sequences C_j:

$$C_j = [\ldots, \tanh(W_j X_{[i:i+l_j-1]} + b_j), \ldots], \quad j \in [1, k]$$

where $X_{[i:i+l_j-1]} = [x_i, x_i+1, \ldots, x_i+l_j-1]$ is the convolutional window with l_j words. After convolution, argument vector is obtained by concatenating maximal value of each sequence:

$$s_j = \max(C_j), \quad R^{(0)} = [s_1, s_2, \ldots, s_k]$$

Long-Short Term Memory Recurrent Neural Network is a variant of RNN and broadly used for modeling sequences. The mechanism in LSTM is showed as follows:

$$
\begin{aligned}
&i_t = \sigma(W_i x_t + U_i c_{t-1} + b_i), \quad f_t = \sigma(W_f x_t + U_f c_{t-1} + b_f), \\
&c_t = f_t \circ c_{t-1} + i_t \circ \tanh(W_c x_t + U_c c_{t-1} + b_c), \\
&o_t = \sigma(W_o x_t + U_o c_t + b_o), \quad h_t = o_t \circ c_t
\end{aligned}
$$

where i_t, f_t and o_t are called input gate, forget gate and output gate. c_t is the cell-state vector that is used to store long-term information. h_t is the output vector.

We get annotations of words by using bidirectional LSTM to summarize information from both directions:

$$
\overrightarrow{h_t} = \overrightarrow{LSTM}(x_t), \quad \overleftarrow{h_t} = \overleftarrow{LSTM}(x_t), \quad h_t = \overrightarrow{h_t} \oplus \overleftarrow{h_t}, \quad t \in [1, L]
$$

Argument vector is then obtained through an attention layer:

$$
\alpha_t = \frac{\exp(h_t^T u)}{\sum \exp(h_t^T u)}, \quad R^{(0)} = \sum \alpha_t h_t
$$

3.3 Mutually Guided Sequence Pairs Modeling

Once guidance vector $Guidance^{(k-1)} \in R^{4D_{k-1}}$ is obtained, we can use it to re-calculate argument vectors. Here, we propose two novel RNN networks with externally controllable cells to implement the strategy of mutually guided modeling. Since the same guidance vector is used for both arguments, the operations for re-calculating new vector representations of the two arguments can not be exactly the same. However, we can share most of parameters among them.

Externally Controllable LSTM (ECLSTM) is derived from LSTM by adding mechanism to enable guidance vector to influence internal gates. In order to maintain conciseness, we omit the (k) and $(k-1)$ superscripts and only take Arg_1 for example.

$$
\begin{aligned}
i_t^1 &= \sigma(W_i x_{1,t} + U_i c_{1,t-1} + V_i^1 Guidance + b_i) \\
f_t^1 &= \sigma(W_f x_{1,t} + U_f c_{1,t-1} + V_f^1 Guidance + b_f) \\
o_t^1 &= \sigma(W_o x_{1,t} + U_o c_{1,t-1} + V_o^1 Guidance + b_o) \\
c_{1,t} &= f_t^1 \circ c_{1,t-1} + i_t^1 \circ \tanh(W_c x_{1,t} + U_c c_{1,t-1} + b_c) \\
h_{1,t} &= o_t^1 \circ c_{1,t}
\end{aligned}
$$

where $W_i, W_f, W_o \in R^{D_k \times D_e}$, $U_i, U_f, U_o \in R^{D_k \times D_k}$, $V_i^1, V_f^1, V_o^1 \in R^{D_k \times 4D_{k-1}}$ and $b_i, b_f, b_o \in R^{D_k}$. The parameters without superscript 1 are shared between processing of Arg_1 and Arg_2. The parameters not shared between two arguments

are those directly multiplied with guidance vector. So, V_i^1, V_f^1 and V_o^1 are used exclusively in processing of Arg_1 and V_i^2, V_f^2 and V_o^2 are used exclusively in processing of Arg_2.

ECLSTM slightly differs from the original LSTM to retain most advantages of LSTM. However, since guidance vector has to interact with inputs and hidden states to generate gates' values, the active mechanism of guidance vector's influence is quit unclear.

Attention-Augmented Gated Recurrent Unit (AAGRU) is proposed because the influence of guidance vector to the gates in ECLSTM is quit limited. We strengthen the dominance of guidance vector by incorporating attention mechanism into GRU that we can reduce parameters in one single layer and stack more guided layers without increasing complexity. Specifically, we sequentially process each word in Arg_i as below:

$$\begin{aligned}
u_i &= \tanh(V_u^i Guidance + b_u) \\
r_t^i &= \sigma(W_r x_{i,t} + U_r c_{i,t-1} + b_r) \\
\tilde{c}_{i,t} &= \tanh(W_c x_{i,t} + U_c(r_t^i \circ c_{i,t-1}) + b_h) \\
a_t^i &= \sigma(\tilde{c}_{i,t}^T u_i) \\
c_{i,t} &= (1 - a_t^i) \circ c_{i,t-1} + a_t^i \circ \tilde{c}_{i,t} \\
h_{i,t} &= a_t^i \circ c_{i,t}
\end{aligned}$$

Here, we still omit the (k) and $(k-1)$ superscripts to maintain conciseness and superscript i and subscript i have the same meaning that the variable is exclusive for $Arg_i, i \in \{1, 2\}$. In AAGRU, the original update gate of GRU is replaced by the attention gate a_t^i. We first generate attention vector u_i for Arg_i, then u_i is used to perform inner product with temporal cell-state vector $\tilde{c}_{i,t}$ to calculate attention gate's value. Obtaining output $h_{i,t}$ by multiplying cell-state vector $c_{i,t}$ with attention gate a_t^i is to alleviate imbalance introduced by the fierce mechanism.

To get the new annotations of words, we summarize information from both sides as conducted in bi-LSTM:

$$\overrightarrow{h_{i,t}} = \overrightarrow{g_i}(x_{i,t}), \quad \overleftarrow{h_{i,t}} = \overleftarrow{g_i}(x_{i,t}), \quad h_{i,t} = \overrightarrow{h_{i,t}} \oplus \overleftarrow{h_{i,t}}, \quad t \in [1, L_i]$$

Here g_i represents $ECLSTM_i$ and $AAGRU_i$. Then we get argument vectors by simply summing word vectors:

$$R_i = \frac{1}{L_i} \sum_t h_{i,t}, \quad R_i \in R^{2D_k}.$$

3.4 Model Training

After the hierarchical architecture of our model is detailed and implemented, we train our model to minimize the cross-entropy error between y the golden

standard relations and P the outputs of the softmax layer as well as the L2 regularization of arguments:

$$L(\theta) = \frac{1}{m}\sum_{k=1}^{m}\big(-\sum_{j} y_{kj}\log P_{kj}\big) + \frac{\lambda}{2}\theta^T\theta$$

Dropout operations are applied between each layers and the keeping ratio is set to be 0.5. The model is trained end to end through standard back-propagation. We adopt AdamOptimizer [14] with initial learning rate of 0.001 for optimization process.

4 Experiments

4.1 Settings

Penn Discourse Treebank (PDTB) is a manually annotated corpus. We evaluate our model on this corpus. The classification granularity of PDTB has three levels and the first level contains four kinds of relations, namely Comparison, Contingency, Expansion and Temporal. To compare with prior works, we follow traditions that formulate the implicit discourse classification tasks as four one-versus-other binary classification problems. Data is divided into three sets, respectively training set (sections 2-20), validation set (sections 0-1) and testing set (sections 21-22). Since positive samples in Temporal are quit limited, we augment training data of Temporal by exchanging the positions of Arg_1 and Arg_2 in it.

We use Stanford NLP Toolkit [9] to conduct tokenization, lemmatization and part-of-speech tagging. The embedding vectors of lemmas are initialized with pre-trained vectors provide by Glove [10]. To prevent from overfitting, in model training, we do not update lemma embedding vectors in the first ten epochs and then use validation set to conduct early stopping.

The dimensions of lemma embeddings and POS embeddings are 300 and 50 respectively. When using CNN as the general feature extractor, we utilize three groups of filters with window sizes of (2, 4, 8) and their filter numbers are all set to 256. So the dimension of general level's output is 768. When using LSTM as the general modeling method, D_0 the dimension of each unidirectional LSTM is set to 50 that the dimension of general level's output is 100. The dimensions of next two guided levels' outputs are set to 100, which means D_k the output size of the unidirectional RNN variants is 50.

4.2 Results

We set our model with different stacked layers and components and their performances are compared in Table 1.

The basic CNN and LSTM models perform the worst. By stacking guided layers, we get significant performance gains. We also witness that stacking more ECLSTM degenerates performance. It may be because ECLSTM contains a lot

Table 1: Models with variable layers measured by F_1 scores (%).

	Comparison	Contingency	Expansion	Temporal
CNN Only	33.05	50.92	63.51	28.42
CNN+ECLSTM	39.33	54.71	69.51	**33.44**
CNN+AAGRU	38.23	52.41	65.07	29.05
CNN+ECLSTM+ECLSTM	38.47	52.65	66.99	28.59
CNN+AAGRU+AAGRU	38.69	55.47	69.41	32.40
LSTM Only	36.11	53.04	67.23	27.74
LSTM+ECLSTM	39.08	53.97	68.33	32.49
LSTM+AAGRU	38.77	53.35	66.81	30.14
LSTM+ECLSTM+ECLSTM	37.39	54.08	67.89	27.42
LSTM+AAGRU+AAGRU	**40.03**	**56.38**	**70.10**	32.85

of parameters that it begin to overfit. AAGRU with reinforced controlling power and less parameters can continually improve performance by stacking more layers. In our experiments, we find that the model with LSTM as general level and two stacked AAGRU as guided levels achieves the best over-all performance.

Table 2: Comparisons of different models measured by F_1 scores (%).

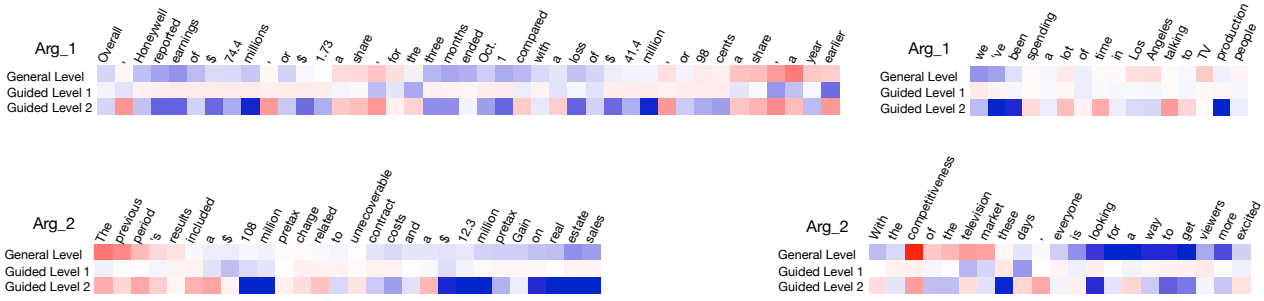
	Comparison	Contingency	Expansion	Temporal
Zhou et al., 2010	31.79	47.16	70.11	20.30
Park and Cardie, 2012	31.32	49.82	-	26.57
Rutherford and Xue, 2014	39.70	54.42	70.23	28.69
Braud and Denis, 2015	36.36	55.76	67.42	29.3
Zhang et al., 2015	34.22	52.04	69.59	30.54
Wu et al., 2016	37.07	42.37	66.84	23.81
Liu et al., 2016	37.91	55.88	69.97	**37.17**
Qin et al., 2017	**40.87**	54.56	**72.38**	36.20
CNN+ECLSTM	39.33	54.71	69.51	33.44
LSTM+AAGRU+AAGRU	40.03	**56.38**	70.10	32.85

We compare our models with previously state-of-the-art models and the results are listed in Table 2. The main ideas of those baselines are introduced as follows:

- (Zhou et al., 2010) [6] inserts discourse connectives between arguments with the use of a language model and use these predicted implicit connectives as additional features in a supervised model.
- (Park and Cardie, 2012) [7] provides a systematic study of linguistic features and identifies new feature combinations that optimize F1-score.
- (Rutherford and Xue, 2014) [8] employs Brown cluster pairs and coreference patterns as features and trains a maximum entropy classifier.

– (Braud and Denis, 2015) [12] uses low-dimensional representations based on Brown clusters and word embeddings and other hand-crafted features to identify implicit discourse relations.
– (Zhang et al., 2015) [13] proposes a shallow convolutional neural network, which contains only one hidden layer and captures max, min and average information.
– (Wu et al., 2016) [16] incorporates data from corpus with different languages via a multi-task neural network model to alleviate the shortage of labeled data.
– (Liu et al., 2016) [18] proposes a convolutional neural network embedded multi-task learning system to combine different discourse corpora.
– (Qin et al., 2017) [20] proposes an adversarial network, which leverages implicit connectives manually added in PDTB, as an regularization mechanism to adaptively regularize parameters.

From the comparison results, we can conclude that our model is better in recognizing Comparison and Contingency and comparative with the best models in classifying Expansion and Temporal.



(a) Comparison Example (b) Contigency Example

Fig. 2: Visualization of Intermediate Weights. (a) is an example of the Comparison relation and (b) is an example of Contingency. Each one contains two figures for its two arguments respectively. Each figure illustrate three layers of weights that correspond to the attention weights of general level and the values of attention gates in the next two AAGRU layers.

4.3 Discussion

We have quantitatively analyzed the attention weights in the general level and the values of attention gates in the guided levels of the LSTM+AAGRU+AAGRU model. Some examples are visualized in Figure 2. More blue the grid means the word is paid less attention than average, while more red means more attention is paid. We find that general level focuses on proper nouns, terminologies and time-related phrases, which are generally most informative components for getting the general sense of a text. The first guided level tends to focus on components which are neglected by general level and can be seen as a complement of

general level. The second guided level, which is also the final level to generate argument representations, is relatively similar to general level. It is more accurate and comprehensive than the general level that it can be seen as a refinement of general level. It discards components which are paid attention by general level but inessential for classifying the relation and allocates extra attention to those verbs and adjectives which are neglected but important.

5 Conclusion

We incorporate arguments' interactions into their modeling process to dynamically exploit critical information for implicit discourse classification. We designed ECLSTM and AAGRU, two variants of recurrent neural network units with externally controllable gating mechanism, to regenerate argument representations under the guidance of previous obtained argument vectors. Our experiments demonstrate that performance can be greatly promoted by applying the guided regenerating strategy. Due to the shortage of labeled data, we stacked only two AAGRU layers to get the best performance. Through visualization we show that relevant information are gradually complemented and refined layer by layer.

References

1. Marcu, D., Echihabi, A.: An Unsupervised Approach to Recognizing Discourse Relations. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, pp. 368–375 (2002)
2. Prasad, R., Dinesh, N., Lee, A., Miltsakaki, E., Robaldo, L., Joshi, A., Webber, B.: The Penn Discourse Treebank 2.0. Proceedings of the 6th International Conference on Language Resources and Evaluation (2008)
3. Pitler, E., Raghupathy, M., Mehta, H., Nenkova, A., Lee, A., Joshi, A.: Easily Identifiable Discourse Relations. Proceedings of the 22nd International Conference on Computational Linguistics (2008)
4. Pitler, E., Louis, A., Nenkova, A.: Automatic Sense Prediction for Implicit Discourse Relations in Text. 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, pp. 683–691 (2009)
5. Lin, Z., Ng, H.T., Kan, M.Y.: PDTB-Styled End-to-End Discourse Parser. Technical report, School of Computing, National University of Singapore (2010)
6. Zhou, Z., Lan, M., Xu, Y., Niu, Z., Su, J., T, C.L.: Predicting Discourse Connectives for Implicit Discourse Relation Recognition. Proceedings of the 23rd International Conference on Computational Linguistics, pp. 1507–1514 (2010)
7. Park, J., Cardie, C.: Improving Implicit Discourse Relation Recognition Through Feature Set Optimization. Proceedings of the 13th Annual Meeting of the Special Interest Group on Discourse and Dialogue, pp. 108–112 (2012)
8. Rutherford, A.T., Xue, N.: Discovering Implicit Discourse Relations Through Brown Cluster Pair Representation and Coreference Patterns. Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pp. 645–654 (2014)

9. Manning, C.D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S.J., McClosky, D.: The Stanford CoreNLP Natural Language Processing Toolkit. Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations, pp. 55–60 (2014)
10. Pennington, J., Socher, R., Manning, C.D.: GloVe: Global Vectors for Word Representation. Proceedings of the 2014 Conference Empiricial Methods in Natural Language Processing (2014)
11. Wang, J., Lan, M.: A Refined End-to-End Discourse Parser. Proceedings of the Nineteenth Conference on Computational Natural Language Learning: Shared Task, pp. 17–24 (2015)
12. Braud, C., Denis, P.: Comparing Word Representations for Implicit Discourse Relation Classification. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2201–2211, (2015)
13. Zhang, B., Su, J., Xiong, D., Lu, Y., Duan, H., Yao, J.: Shallow Convolutional Neural Network for Implicit Discourse Relation Recognition. Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, pp. 2230–2235 (2015)
14. Kingma, D.P., Adam, J.B.: A Method for Stochastic Optimization. Proceedings of the 3rd International Conference for Learning Representations (2015)
15. Chen, J., Zhang, Q., Liu, P., Qiu, X., Huang, X.: Implicit Discourse Relation Detection via a Deep Architecture with Gated Relevance Network. Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, pp. 1726–1735 (2016)
16. Wu, C., Shi, X., Cheng, Y., Huang, Y., Su, J.: Bilingually-constrained Synthetic Data for Implicit Discourse Relation Recognition. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 2306–2312 (2016)
17. Yang, Z., Yang, D., Dyer, C., He, X., Smola, A., Hovy, E.: Hierarchical Attention Networks for Document Classification. Proceedings of NAACL Conference (2016)
18. Liu, Y., Li, S., Zhang, X., Sui, Z.: Implicit Discourse Relation Classification via Multi-Task Neural Networks. Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, pp. 2750–2756 (2016)
19. Liu, Y., Li, S.: Recognizing Implicit Discourse Relations via Repeated Reading: Neural Networks with Multi-Level Attention. Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, pp. 1224–1233 (2016)
20. Qin, L., Zhang, Z., Zhao, H., Hu, Z., Xing, E.P.: Adversarial Connective-exploiting Networks for Implicit Discourse Relation Classification. Proceedings of ACL Conference (2017)
21. Li, H., Zhang, J., Zong, C.: Implicit Discourse Relation Recognition for English and Chinese with Multiview Modeling and Effective Representation Learning. ACM Transactions on Asian and Low-Resource Language Information Processing, vol. 16 (2017)