Using WHY-type Question-Answer Pairs to Improve Implicit Causal Relation Recognition

Huibin Ruan, Yu Hong, Yu Sun, Yang Xu, Min Zhang
School of Computer Science and Technology, Soochow University
Suzhou, China
{*hbruan,ysun79,minzhang*}*@stu.suda.edu.cn*,{*tianxianer,andreaxu41*}*@gmail.com*

***Abstract*—Implicit causal relation recognition aims to identify the causal relation between a pair of arguments. It is a challenging task due to the lack of conjunctions and the shortage of labeled data. In order to improve the identification performance, we come up with an approach to expand the training dataset. On the basis of the hypothesis that there inherently exists causal relations in WHY-type Question-Answer (QA) pairs, we utilize WHY-type QA pairs for the training set expansion. In practice, we first collect WHY-type QA pairs from the Knowledge Bases (KBs) of the reading comprehension tasks, and then convert them into narrative argument pairs by Question-Statement Conversion (QSC). In order to alleviate redundancy, we use active learning (AL) to select informative samples from the synthetic argument pairs. The sampled synthetic argument pairs are added to the Penn Discourse Treebank (PDTB), and the expanded PDTB is used to retrain the neural network-based classifiers. Experiments show that our method yields a performance gain of 2.42% $F1$-score when AL is used, and 1.61% without using.**

***Keywords*-Implicit causal relation recognition; PDTB; Discourse parsing;**

I. Introduction

Discourse relation recognition is a task to determine the relation between a pair of arguments (abbr., *Arg*). This task is important because it can help for many practical Natural Language Processing (NLP) systems, such as automatic text summarization [1], question answering [2] and conversation [3]. In this task, implicit relation recognition is still a challenge due to the lack of explicit connectives (e.g., "because"). Such overt marker can strongly indicate the relation between two arguments [4].

The distinction between explicit and implicit relations is clearly defined in the Penn Discourse Treebank (PDTB) [5]. As shown in example (1) and (2), the explicit relation in (1) can be easily identified in that the explicit clue "*because*" straightforwardly indicates the causal relation. While the implicit relation in (2) is difficult to be recognized in a large part due to the lack of a connective. In this case, a deep semantic inference between two arguments is indispensable to determine the implicit relation.

(1) *They shredded it simply* **because** *the Georgia-Pacific bid broke the market's recent gloom.*
(*Contingency.Cause.Reason* - wsj_0335)

(2) *This is not the case.* (**because**) *Some diaries simply aren't worth snooping in.*
(*Contingency.Cause.Result* - wsj_0972)

Most recently, neural models are popular in discourse parsing, and the $F1$-score of four-way classification for the four main relation types (*Expansion*, *Contingency*, *Comparison* and *Temporal*) has been increased to 51.06% [6]. As claimed in previous work [7], such classification models can be strengthened further if there is a larger dataset put into use for training. Therefore, data expansion becomes progressively important in this case.

Discourse-oriented data expansion can be boiled down to two aspects: mining inherently-related arguments and labeling the exact relations. Both will be labor-intensive and time-consuming if they are left to be done by human. To solve this problem, we propose an approach to expand PDTB with less human intervention.

As naturally related sentence pairs, we extract QA pairs from QA KBs for PDTB expansion. In practice, we convert a question into a declarative sentence, and combine it with its answer to form a pair of arguments. As shown in (3), a causally-related argument pair can be generated by a WHY-type QA pair.

(3) **Question**: *Why is efficiency sometimes lost in phosphor-based LEDs?*
Answer: *heat loss from the Stokes shift*
Question type: WHY

Arg1: "*efficiency is sometimes lost in phosphor-based LEDs*". $_{Result}$
Arg2: "*heat loss from the Stokes shift*". $_{Cause}$
Relation type: *Contingency.Cause.Reason*

In addition, we follow Xu et al. [8] to apply active learning for redundancy elimination. Entropy based informativeness measurement is used to verify whether a pair of newly generated arguments is informative or redundant for learning. The goal is to facilitate the joint use of multiple-source data.

As a preliminary study to evaluate the feasibility of the expansion method mentioned above, we focus on implicit causal relation recognition. So only WHY-type QA pairs are utilized for data expansion [9]. But it is noteworthy that one may take the proposed approach as a baseline to carry out a brand new study of data expansion for multiple-class distant supervision, such as that on temporal and conditional relations other than causal ones, using WHEN-type and HOW-type QA pairs. Experimental results on PDTB show that our approach effectively improves the causal classification performance with in-domain QA pairs, improving the $F1$-score by 2.42%. Besides, active learning is proven effective for redundancy elimination.

978-1-7281-5014-7/19/$31.00 ©2019 IEEE

II. Related Work

In much previous work of causal relation recognition, metrics and traditional machine learning models are utilized for causality classification, such as distributional similarity [10] and rule-based approaches [11].

Neural network-based methods for implicit relation recognition have been proven effective [7], [12], [13], [6], [14], which require a large number of training data. Due to the data shortage problem in PDTB, discourse-oriented data expansion captures the interest of the research community. Therefore, many corpora are utilized for implicit data expansion, such as the explicit data in PDTB [15], [16], [8], FBIS and HongKong Law [7]. Using the expanded PDTB for classifier training helps strengthen relation recognition. Nevertheless, in such corpora, the semantic relations or homogeneous properties actually have been manually annotated more or less. Thus, the expansion methods fail to be migrated to other kinds of corpora freely and compatibly. The difficulty of obtaining a large training dataset cannot be overcome completely.

III. Our Approach

Our approach mainly consists of three parts: the neural network-based classifier, Question-Statement Conversion (QSC) and Active Learning (AL) mechanism (See Fig. 1). Utilizing the benchmark dataset in PDTB for training, we first obtain a well-trained neural classifier. Based on QSC, the counterfeit causally-related argument pairs are converted from WHY-type QA pairs. In order to eliminate redundancy, the synthetic argument pairs are iteratively sampled by AL mechanism and added to the benchmark dataset. On the basis of the expanded training dataset, we retrain and redevelop the neural classifier [8].

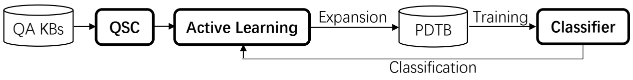

Figure 1: The framework of the proposed approach.

A. Neural Classifier

Basic neural networks akin to Convolutional Neural Network (CNN) and Bi-directional Long Short Term Memory (Bi-LSTM) Network are popular in discourse parsing, which are respectively adopted as classifier in our approach. The reason why we retrain and test them using the expanded datasets is for the purpose of verifying whether our data expansion approach may possibly help to improve other more complicated neural classifier.

B. Question-Statement Conversion

We collect the WHY-type QA pairs and convert them into causally-related argument pairs. QSC is a critical step, which converts a question into a narrative sentence. Such a sentence is then specified as an argument in a role of *result*. Meanwhile the answer accompanied with the question is directly employed as the related argument in a role of *reason*. Therefore, what we necessarily deal with is just to perform interrogation-narration conversion for questions. We first come up with the rule based method for QSC. In order to conduct QSC more automatically, we also utilize Pointer-generator Network [17] for QSC.

Rule based QSC (RQSC): We come up with a series of trivial rules for QSC, most of which can be boiled down to auxiliary verb (AUX) translocation. See the syntax tree shown in Fig. 2, in which the auxiliary is relocated behind the supreme Noun Phrase (NP). The auxiliaries we consider include modal verbs (e.g., "could"), copular verbs (e.g., "was") and regular auxiliaries (e.g., "does").

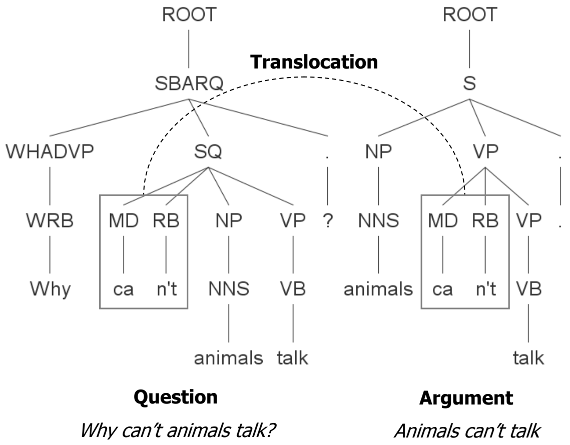

Figure 2: Rule based QSC.

When we conduct QSC for a question, we first parse the sentence using Stanford Parser[1]. Second, we traverse the syntax tree in a depth-first order, seeking for auxiliaries by syntactic roles, such as MD (modality), VBZ (auxiliary in third person singular), etc. We lock onto the first auxiliary we met and the affix if have (e.g., a negative adverb "n't"), and relocate them behind the supreme NP. Finally, we prune the question marker "Why" off the syntax tree. On the basis, we traverse the revised syntax tree in a depth-first order, arrange the words we met in a queue, and perform the dequeue in first-in, first-out order.

In this way, we retroactively produce a declarative sentence, which is used as the argument in the role of "*Result*". The answer of the question is directly used as the argument in the role of "*Reason*".

Pointer-generator network based QSC (PQSC): Pointer-generator network [17] is utilized for QSC to generate a narrative sentence from the given question. It is able to automatically decide whether to generate a word from the given word list, or copy a word from the input question. Given a question $Q = \{x_1, x_2, ..., x_T\}$, each word x_i in it is encoded into an encoder state h_i via a Bi-LSTM. At each decoding step t, the inputs of the decoder are the embedding of the previous word, the previous decoder state s_{t-1} and the context vector h_t^* which is computed by the attention distribution a^t. In practice, a^t

[1] https://nlp.stanford.edu/software/lex-parser.shtml

is computed as follows [18]:

$$e_i^t = v^T tanh(W_h h_i + W_s s_t + b_{attn}) \quad (1)$$

$$a^t = softmax(e^t) \quad (2)$$

where, v, W_h, W_s and b_{attn} are learnable parameters. The distribution a^t is used to compute the context vector h_t^*. And thus, the generation probability $P_\lambda \in [0,1]$ can be estimated using h_t^*, s_t and x_t.

$$h_t^* = \sum_i a_i^t h_i \quad (3)$$

$$P_\lambda = \sigma(w_{h^*}^T h_t^* + w_s^T s_t + w_x^T x_t + b_{ptr}) \quad (4)$$

where, w_{h^*}, w_s, w_x and b_{ptr} are learnable parameters, and σ is the sigmoid function. Acting as a soft switch, P_λ can be used in the determination of whether a word needs to be copied from the input question or generated from a given vocabulary [17]. In practice, it is grounded with a copy distribution $P_c(w)$ over a^t as well as the vocabulary distribution $P_v(w)$. This is implemented by a weighted aggregation function as follows:

$$P(w) = P_\lambda P_v(w) + (1 - P_\lambda) P_c(w) \quad (5)$$

where, $P(w)$ coordinates the probability distributions on the vocabulary and all tokens in the input question.

In order to reduce repetitive during narrative sentence generation, we follow See et al. [17] to maintain a coverage vector c^t and use it to optimize the loss function. Such a vector c^t is used to represent the coverage degree of cumulative attentions over the latent information of a token, thus it is computed as follows:

$$c^t = \sum_{t'=0}^{t-1} a^{t'} \quad (6)$$

where, $a^{t'}$ is a revised version of the attention vector a_t in Eqs. (1) and (2) [17].

We follow See et al. [17] to train the pointer-generator network. The loss consists of two parts: the negative log likelihood [19] of generation probability of the target word w_t^* at time step t and a weighted coverage loss:

$$loss = \frac{1}{T} \sum_{t=0}^{T} \big(- logP(w_t^*) + \gamma \sum_i min(a_i^t, c_i^t) \big) \quad (7)$$

where, γ is used to reweight the coverage loss.

C. Active Learning

To eliminate the noises and redundancy that brought from QSC based data expansion, AL mechanism [8] is employed for informative instances sampling.

Informative samples are those that classified by models with high uncertainty. We use $I_{r_j}(x_i; M)$ to represent the uncertainty level when a sample x_i is recognized as class r_j by a classifier *M* [8]. Thus x_i is informative only if it significantly increases uncertainty:

$$x^* = \arg\max \sum_{r_j \in R} I_{r_j}(x_i; M) \quad (8)$$

We follow Xu et al. [8] to employ entropy-based uncertainty sampling function to measure the informativeness:

$$\begin{aligned} Inf(x_i) &= \sum_{r_j \in R} I_{r_j}(x_i; M) \\ &= - \sum_{r_j \in R} P(r_j \mid x_i) \log P(r_j \mid x_i) \end{aligned} \quad (9)$$

where, the entropy of the probabilities over causal relation is specified as the informativeness score for instance x_i.

AL mechanism [8] used in our approach mainly includes four steps:

- **Step 1**: Train a learning model over the labeled data in PDTB.
- **Step 2**: Use the well-trained model to classify the unlabeled data.
- **Step 3**: Evaluate the informativeness of the unlabeled data. To reduce the computational complexity, the samples with informativeness scores that higher than a threshold 0.95 [8] are selected. They should have be manually annotated, but the relation of each argument pair is already known.
- **Step 4**: Add the samples to the labeled data and retrain the classifier.

In general, the AL iteration process will not stop until the termination criteria are satisfied, such as the upper bound number of iteration is met [8].

Source code to reproduce the experiments will be made publicly available.

IV. Experiments

A. Data Standardization and Expansion

We will first introduce the benchmark dataset of our approach, and then describe the data expansion procedure based on QSC.

Benchmark Dataset: Ponti and Korhonen [20] utilizes *Contingency* argument pairs as positive samples for causal classification, and other types as negative. However, *Contingency* contains 4 subtypes: *Cause*, *Pragmatic Cause*, *Condition* and *Pragmatic Condition*, which are slightly different in relation sense. Especially, there is no causal influence between the arguments of a *Pragmatic Cause* relation [21]. To conduct causal classification in a pure environment, only *Cause* argument pairs are used as positive samples in our work, which is a subset of the Ponti and Korhonen's benchmark dataset (**Sub-P&K's BD** for short).

Table I: Data distribution of Sub-P&K's BD.

Datasets	Train	Dev	Test	*all*
Positive	3,277	284	272	3,833
Negative	3,277	899	774	4,950

In our experiments, the benchmark dataset Sub-P&K's BD is separated into training, dev and test sets. Table I describes the data distribution after the numbers of positive and negative samples are balanced in the training set [22].

Table II: Causal classification performances of the retrained models. (Train: training set for the pointer-generator network. "-" denotes using RQSC (except for the benchmark dataset Sub-P&K's BD); Estimate: training set for classifiers. In "Estimate" column, the QA KBs are those which used for Sub-P&K's BD expansion.)

Datasets		CNN				Bi-LSTM			
Train	Estimate	P(%)	R(%)	F(%)	Gain(%)	P(%)	R(%)	F(%)	Gain(%)
-	Sub-P&K's BD	36.16	79.77	49.77	-	34.74	81.62	48.74	-
-	+SQuAD	**39.08**	**75.00**	**51.38**	**↑1.61**	**41.50**	**61.03**	**49.40**	**↑0.66**
-	+MARCO	35.79	83.82	50.16	↑0.39	40.35	59.93	48.22	↓0.52
-	+NarrativeQA	34.23	79.41	47.84	↓1.93	34.61	76.47	47.65	↓1.09
MARCO	+SQuAD	**40.04**	**70.95**	**51.19**	**↑1.42**	36.41	65.07	46.70	↓2.04
NarrativeQA	+SQuAD	42.82	61.39	50.45	↑0.68	34.06	73.90	46.64	↓2.11
SQuAD	+MARCO	38.59	68.38	49.34	↓0.43	35.40	65.07	45.85	↓2.89
NarrativeQA	+MARCO	36.89	81.25	50.75	↑0.98	34.58	71.32	46.58	↓2.16
MARCO	+NarrativeQA	34.65	87.13	49.58	↓0.19	36.47	66.91	47.21	↓1.53
SQuAD	+NarrativeQA	36.93	76.84	49.88	↑0.11	38.54	56.25	45.74	↓3.00

Data Expansion: On the basis of QSC, we convert a question into a narrative sentence, which serves as a *"Result"* in the synthetic causal argument pair. Meanwhile, the answer of the question directly serves as a *"Reason"*. With these causally-related argument pairs, we expand the training set in the benchmark dataset. The following three QA KBs are used for expansion:

- **SQuAD** [23]: contains 107,785 QA pairs generated from 536 Wikipedia articles by trained annotators. Based on the answer spans, the questions are generated editorially. We extract 1,028 WHY-type QA pairs for use.
- **MARCO** [24]: consists of 100,000 queries issued to the *Bing* search engine by real users, and the corresponding answers are also free-form human generated text. There are 1,298 WHY-type QA pairs identified and taken for use.
- **NarrativeQA** [25]: includes 46,765 QA pairs generated from 1,572 stories by trained annotators. There are 4,015 WHY-type QA pairs utilized.

Based on the datasets mentioned above, we obtain three sets of synthetic argument pairs by RQSC. Therefore, three expanded versions of the benchmark dataset are generated by adding them to the Sub-P&K's BD respectively. While for the PQSC, we first train the pointer-generator network on one of the QA KBs. Using the well-trained model, we generate synthetic causal argument pairs for another QA KB. The counterfeit argument pairs are added to the Sub-P&K's BD to form an expanded version of the benchmark dataset. Running the procedure mentioned above for six times, we obtain six expanded versions of the Sub-P&K's BD by PQSC.

Totally, we obtain nine expanded versions of the Sub-P&K's BD by QSC. In the training set of each expanded Sub-P&K's BD, the number of positive samples is much higher than the negative ones. This leads to unbalance between positive and negative samples, which has been widely recognized as one of the reasons for performance reduction [26]. Therefore, we randomly select negative samples from PDTB to maintain the balance.

B. Experimental Setup

Table III: Hyperparameter settings for neural models.

Hyperparameter	CNN	Bi-LSTM	Pointer-generator network
Learning rate	$1e-3$	0.01	0.1
Batch size	64	30	16
Optimizer	Adam	Adam	Adagrad
Dropout rate	0.2	0.1	-
Filters number	$1,024$	-	-
Filter size	$(2,2,2)$	-	-

We respectively train the neural classifiers on Sub-P&K's BD and the nine expanded versions of it (See Section IV-A), and adopt Precision (P), Recall (R) and $F1$-score as the evaluation metrics. The hyperparameter settings are shown in Table III and the detailed model settings are as follows:

CNN [22] is used to recognize the causal relation. We combine word embeddings and POS embeddings to represent the arguments. The 300-dimensional word embeddings are initialized with pre-trained Word2Vec [27] vectors, and the 50-dimensional POS embeddings are initialized by random sampling in [-1,1].

Bi-LSTM [26] is adopted for argument modelling and relation recognition. We follow Guo et al. [26] to set the max sentence length as 50. The 50-dimensional word embeddings are initialized with GloVe [28] vectors. The size of the hidden state for LSTM is set to 100.

Pointer-generator network [17] is adopted for QSC. Adagrad [29] is used to optimize the learnable parameters. During training, the learning rate is set to 0.1 and the initial accumulator value is set as 0.1. The decoding beam size is set to 4 and the coverage loss weight γ is set to 1. We follow See et al. [17] to employ a vocabulary that is extracted from the training sets, and the size is limited to 50k. For both the encoder and decoder, we use 256-dimensional hidden states and 300-dimensional word embeddings initialized with Word2Vec [27] vectors.

C. Experimental Results and Analysis

As shown in Table II, when training on the Sub-P&K's BD expanded by 1,028 synthetic causal argument pairs generated from SQuAD, the $F1$-scores are improved by 1.61% and 0.66% on CNN and Bi-LSTM respectively. However, utilizing QA pairs in NarrativeQA for expansion hurts the performance, thought there are 4,015 WHY-type QA pairs used. Undoubtedly, this raises an adaptation

Table IV: Causal classification performances of the retrained models when AL is used.

Datasets		CNN				Bi-LSTM			
Train	Estimate	P(%)	R(%)	F(%)	Gain(%)	P(%)	R(%)	F(%)	Gain(%)
-	Sub-P&K's BD	36.16	79.77	49.77	-	34.74	81.62	48.74	-
-	+SQuAD(AL)	**41.52**	**70.22**	**52.19**	**↑2.42**	**39.35**	**71.32**	**50.72**	**↑1.98**
-	+MARCO(AL)	39.50	76.10	52.01	↑2.24	37.55	72.06	49.37	↑0.63
-	+NarrativeQA(AL)	41.50	67.28	51.33	↑1.56	36.69	75.00	49.28	↑0.54
MARCO	+SQuAD(AL)	**43.03**	**65.81**	**52.03**	**↑2.26**	**36.26**	**80.51**	**50.00**	**↑1.26**
NarrativeQA	+SQuAD(AL)	40.63	71.69	51.86	↑2.09	36.69	75.00	49.28	↑0.54
SQuAD	+MARCO(AL)	40.39	68.01	50.68	↑0.91	37.17	73.53	49.38	↑0.64
NarrativeQA	+MARCO(AL)	38.71	75.00	51.06	↑1.29	35.43	74.63	48.05	↓0.69
MARCO	+NarrativeQA(AL)	40.08	71.32	51.32	↑1.55	37.88	69.49	49.03	↑0.29
SQuAD	+NarrativeQA(AL)	41.90	64.71	50.87	↑1.10	38.0	69.85	49.29	↑0.55

problem. Our survey illustrates that the inconsistency between domains most probably results in less adaptability.

The experimental results show that the domain inconsistency probably leads to less adaptability. The arguments in PDTB are manually extracted from news stories of the Wall Street Journal. While the QA KBs are established on encyclopedias, webpages, movie scripts and books. Thus the argument pairs in PDTB is rhetorically closer to those in SQuAD. This makes it easier for neural classifiers to learn isomorphic linguistic knowledge from SQuAD samples. By contrast, the rhetorically heterogeneous pragmatics are widely distributed in the NarrativeQA samples. Therefore, data expansion using these samples easily introduces unintelligible latent features for learning rather than referential ones. This can be illustrated with some representative samples akin to those in (4) and (5), where the PDTB argument pair in (4) are rhetorically dissimilar to the QA pair of NarrativeQA in (5). Note that they have the same topic of "*divulging*", though they are expressed in different rhetoric.

(4) *Arg pair* in **PDTB**: [*One of them, 25-year-old Markus Hess of Hannover, allegedly used the international telecommunications network to break into more than 30 high-security computers in the U.S., searching for secrets.*]$_{Arg1}$ [*He probably didn't penetrate any top-secret files, but the KGB in East Berlin was willing to pay two of his associates.*]$_{Arg2}$

(5) *QA pair* in **NarrativeQA**: [*Why does Harry reveal his secret life?*]$_{Question}$ [*He is given a truth serum.*]$_{Answer}$

Obviously, the retrained Bi-LSTM performs worse than CNN when the PQSC is used for expansion. Meanwhile, the classifiers perform slightly better when using the RQSC for expansion. These mainly result from that the synthetic argument pairs generated by PQSC include some less-ordered cases (See (6)).

(6) **Question**: *Why was their use limited?*
Ground Truth: *their use was limited.*
Generated by PQSC: *as use limited was their.*

D. Discussion

In order to eliminate redundancy, we follow Xu et al. [8] to use active learning mechanism to purify the hold-up set for data expansion. As shown in Table IV, after using AL mechanism to select informative samples for expansion, almost all neural classifiers have been improved further and performs better than only using the Sub-P&K's BD. In particular, the performance gains of CNN and Bi-LSTM have been increased to 2.42% and 1.98% respectively when the Sub-P&K's BD is expanded with SQuAD. The experimental results indicate that the classifiers are more adaptable, when the training set is expanded with samples selected by AL mechanism from various external datasets.

Table V: Causal classification performances of the retrained models when the benchmark dataset is expanded with samples selected by AL from different data sources.

Expansion Source	*num* (AL)/ALL	P(%)	R(%)	F(%)
Xu et al. [8]	291/2,240	44.79	58.46	50.72
Ours	269/1,028	**41.52**	**70.22**	**52.19**

In order to prove that using in-domain QA pairs to expand the benchmark dataset enhances the neural classifiers, we compare our method with the state-of-the-art AL based expansion approach [8] for implicit causal relation recognition. As show in Table V, our approach performs better than Xu et al. [8]'s when using the same neural classifier, AL algorithm and corpus (i.e., PDTB). While Xu et al. [8] use 2,240 explicit causal argument pairs in PDTB for expansion, and we use 1,028 QA pairs in SQuAD. With less candidates, we select 269 informative samples for expansion, almost the same number as Xu et al. [8] obtain.

V. Conclusion

We propose to use WHY-type QA pairs for data expansion, which enhances the performance of implicit causal relation recognition. In addition, we utilize active learning mechanism for redundancy elimination. In the future, we will further explore the usage of HOW-type and WHEN-type QA pairs, so as to help relation *Condition* and *Temporal* recognition.

Acknowledgment

This research work is supported by National Natural Science Foundation of China (Grants No.61672367, No.61672368, No.61751206.). The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. Yu Hong, Professor in Soochow University, is the corresponding author of the paper, whose email address is tianxianer@gmail.com.

References

[1] Y. Yoshida, J. Suzuki, T. Hirao, and M. Nagata, "Dependency-based discourse parser for single-document summarization," in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1834–1839.

[2] P. Jansen, M. Surdeanu, and P. Clark, "Discourse complements lexical semantics for non-factoid answer reranking," in *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, vol. 1, 2014, pp. 977–986.

[3] R. Higashinaka, K. Imamura, T. Meguro, C. Miyazaki, N. Kobayashi, H. Sugiyama, T. Hirano, T. Makino, and Y. Matsuo, "Towards an open-domain conversational system fully based on natural language processing," in *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014, pp. 928–939.

[4] E. Pitler, A. Louis, and A. Nenkova, "Automatic sense prediction for implicit discourse relations in text," in *ACL*. Association for Computational Linguistics, 2009, pp. 683–691.

[5] R. Prasad, N. Dinesh, A. Lee, E. Miltsakaki, L. Robaldo, A. Joshi, and B. Webber, "The penn discourse treebank 2.0," in *LREC*, 2008.

[6] H. Bai and H. Zhao, "Deep enhanced representation for implicit discourse relation recognition," in *COLING*, 2018, pp. 571–583.

[7] C. Wu, Y. Chen, Y. Huang *et al.*, "Bilingually-constrained synthetic data for implicit discourse relation recognition," in *EMNLP*, 2016, pp. 2306–2312.

[8] Y. Xu, Y. Hong, H. Ruan, J. Yao, M. Zhang, and G. Zhou, "Using active learning to expand training data for implicit discourse relation recognition," in *EMNLP*, 2018, pp. 725–731.

[9] Z. Lin, H. T. Ng, and M.-Y. Kan, "A pdtb-styled end-to-end discourse parser," *Natural Language Engineering*, vol. 20, no. 2, pp. 151–184, 2014.

[10] Q. X. Do, Y. S. Chan, and D. Roth, "Minimally supervised event causality identification," in *EMNLP*. Association for Computational Linguistics, 2011, pp. 294–303.

[11] C. Grivaz, "Automatic extraction of causal knowledge from natural language texts," Ph.D. dissertation, University of Geneva, 2012.

[12] P. Liu, X. Qiu, and X. Huang, "Adversarial multi-task learning for text classification," *arXiv preprint arXiv:1704.05742*, 2017.

[13] M. Lan, J. Wang, Y. Wu, Z.-Y. Niu, and H. Wang, "Multi-task attention-based neural networks for implicit discourse relationship representation and identification," in *EMNLP*, 2017, pp. 1299–1308.

[14] Z. Dai and R. Huang, "Improving implicit discourse relation classification by modeling inter-dependencies of discourse units in a paragraph," in *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018, pp. 141–151.

[15] D. Marcu and A. Echihabi, "An unsupervised approach to recognizing discourse relations," in *ACL*. Association for Computational Linguistics, 2002, pp. 368–375.

[16] A. Rutherford and N. Xue, "Improving the inference of implicit discourse relations via classifying explicit discourse connectives." in *HLT-NAACL*, 2015, pp. 799–808.

[17] A. See, P. J. Liu, and C. D. Manning, "Get to the point: Summarization with pointer-generator networks," *arXiv preprint arXiv:1704.04368*, 2017.

[18] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[19] F. Qian, C. Gong, K. Sharma, and Y. Liu, "Neural user response generator: Fake news detection with collective user intelligence." in *IJCAI*, 2018, pp. 3834–3840.

[20] E. M. Ponti and A. Korhonen, "Event-related features in feedforward neural networks contribute to identifying causal relations in discourse," *LSDSem 2017*, p. 25, 2017.

[21] R. Prasad, E. Miltsakaki, N. Dinesh, A. Lee, A. Joshi, L. Robaldo, and B. L. Webber, "The penn discourse treebank 2.0 annotation manual," 2007.

[22] L. Qin, Z. Zhang, and H. Zhao, "A stacking gated neural architecture for implicit discourse relation classification," in *EMNLP*, 2016, pp. 2263–2270.

[23] P. Rajpurkar, J. Zhang, K. Lopyrev, and P. Liang, "Squad: 100,000+ questions for machine comprehension of text," *arXiv preprint arXiv:1606.05250*, 2016.

[24] T. Nguyen, M. Rosenberg, X. Song, J. Gao, S. Tiwary, R. Majumder, and L. Deng, "Ms marco: A human generated machine reading comprehension dataset," *arXiv preprint arXiv:1611.09268*, 2016.

[25] T. Kočiskỳ, J. Schwarz, P. Blunsom, C. Dyer, K. M. Hermann, G. Melis, and E. Grefenstette, "The narrativeqa reading comprehension challenge," *arXiv preprint arXiv:1712.07040*, 2017.

[26] F. Guo, R. He, D. Jin, J. Dang, L. Wang, and X. Li, "Implicit discourse relation recognition using neural tensor network with interactive attention and sparse learning," in *COLING*, 2018, pp. 547–558.

[27] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *arXiv preprint arXiv:1301.3781*, 2013.

[28] J. Pennington, R. Socher, and C. Manning, "Glove: Global vectors for word representation," in *EMNLP*, 2014, pp. 1532–1543.

[29] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.