DPR at SemEval-2021 Task 8: Dynamic Path Reasoning for Measurement Relation Extraction

Amir Pouran Ben Veyseh[1], **Franck Dernoncourt**[2],
and **Thien Huu Nguyen**[1]
[1] Department of Computer and Information Science, University of Oregon,
Eugene, OR 97403, USA
[2] Adobe Research, San Jose, CA, USA
{apouranb,thien}@cs.uoregon.edu,
franck.dernoncourt@adobe.com

Abstract

Scientific documents are replete with measurements mentioned in various formats and styles. As such, in a document with multiple quantities and measured entities, the task of associating each quantity to its corresponding measured entity is challenging. Thus, it is necessary to have a method to efficiently extract all measurements and attributes related to them. To this end, in this paper, we propose a novel model for the task of measurement relation extraction (MRE) whose goal is to recognize the relation between measured entities, quantities and conditions mentioned in a document. Our model employs a deep translation-based architecture to dynamically induce the important words in the document to classify the relation between a pair of entities. Furthermore, we introduce a novel regularization technique based on Information Bottleneck (IB) to filter out the noisy information from the induced set of important words. Our experiments on the recent SemEval 2021 Task 8 datasets reveal the effectiveness of the proposed model.

1 Introduction

One of the key indicators of scientific writing is the quantities description of various experiments and results. While the mentions of all measurements could provide a rigorous understanding of the topic, it might make the reading and automatic processing of the text more difficult. As such, designing effective methods to recognize the mentions of measurements and also the conditions in which they are valid is necessary. According to the definition of the SemEval 2021 Task 8 (Harper et al., 2021), a measurement might consist of the following components: (i) Measure Entity: A span referring to an entity that one of its properties has been measured and its value is provided in the document; (ii) Measured Property: A span referring to the characteristics of an entity that has been measured; (iii) Quantity: A span in the document that refers to a value and possibly it comes with a unit; and (iv) Qualifier: A span referring to a condition in which more information about the Quantity, Measured Property or Measured Entity is provided. Figure 1 shows a sample document annotated with the aforementioned entities. In this paper, we collectively name all of these four types as measurement component.

[ME1] samples [/ME1] have been generated with Coronin and Dystrophin proteins. In the filtration experiments, some of them with a [PR1] diameter[/PR1] [QT1] less than 2 mm [/QT1] have been filtered out using [QT2] 200-degree [/QT2] filtering [ME2] radiation [/ME2], resulting in [QT3] 20% [/QT3] [ME3] utilization [/ME3]. These results are obtained in a [QL1] dry climate [/QL1].

Figure 1: A document annotated with the measured entities (i.e., [ME]), quantity (i.e., [QT]), measured property (i.e., [PR]) and qualifier (i.e., [QL]) (best viewed in color).

As it is shown in the provided example, documents might contain multiple entities, properties, quantities and qualifiers that are scattered in different parts of the document. As such, finding which measurement components are associated with each other is not straightforward. In this paper, this task is called measurement relation extraction (MRE) that aims to recognize what is the relationship between two given measurement components. More specifically, the following relation types are considered: (i) Has-property: Indicates the selected property is one of the characteristics of the selected entity; (ii) Has-Quantity: Indicates the selected quantity is provided for the selected entity or property; (iii) Qualifies: Indicates the selected qualifier provides more information about the selected entity or quantity; (iv) None: Indicates that there is no

Proceedings of the 15th International Workshop on Semantic Evaluation (SemEval-2021), pages 397–403
Bangkok, Thailand (online), August 5–6, 2021.

relation between the selected measurement components. For instance, in the given example document in Figure 1, the following relations between different measurement components exist: (1) ME_1 *has property* PR_1; (2) PR_1 *has quantity* QT_1; (3) ME_2 *has quantity* QT_2; (4) ME_3 *has quantity* QT_3; and (5) QL_1 *qualifies* ME_3;

Finding the relation between a pair of measurement components is challenging and it requires consideration about the position of the given entities and the context in which they are used. Generally, this task can be formulated as a typical Relation Extraction (RE) task whose goal is to identify the semantic relation between two given named entity mentions. For RE, it has been shown that contextual information such as dependency path between the two given entities is important. As such, in this paper, we also aim to exploit the contextual information for a pair of measurement entities to predict the relation between them. To this end, the main question to answer is how we can extract the contextual information that is helpful for this task. One simple solution is to use the dependency path between the two measurement components. However, this might not be perfect due to various reasons such as lack of high-quality dependency parser designed especially for scientific domain and the fact that the dependency tree is ignorant of the downstream task (i.e., MRE) thus might not be efficient to extract important context from. Therefore, in this paper, we aim to propose a novel method to dynamically infer the important context for the MRE task. More specifically, we introduce a deep architecture to infer which words should be selected from the given document to form the important context from which the relation between the given measurement components can be inferred. The proposed deep architecture exploits a translation-based perspective to achieve this goal.

In addition, in this paper, we propose a novel method to efficiently regularize the representations of the input words based on the inferred important context. In particular, our method is based on the Information Bottleneck (IB) theory in which the inferred context is treated as information bottleneck to exclude noisy information in the input document representation. We conduct extensive experiments on the SemEval 2021 Task 8 dataset. Our experiments reveal the effectiveness of the proposed model for the task of MRE.

2 Model

Task Definition: The input to the model is the document $D = [w_1, w_2, \ldots, w_n]$ consisting of n words and also the positions of the two entities of interest, w_s and w_o where s and o are the indices of the first (i.e., subject) and the second (i.e., object) entities, respectively. The input document is annotated with the label l from the set $L = \{\text{hasQuantity}, \text{hasProperty}, \text{qualifies}, \text{None}\}$. Our proposed model for this task consists of four major components: (1) Input encoder to convert the input text into high dimensional word vectors; (2) Dependency Path Reasoning: This component employs the word vector representations and extract a path between the two entity mentions in the given document; (3) Regularization: This component employs the extracted dependency path as the information bottleneck to filter out noisy information from the input document; (4) Prediction: Finally the regularized representations of the dependency path will be used to make the final prediction. The rest of this section provides details for the aforementioned components.

2.1 Input Encoder

To represent each word w_i in the input document D, we use the concatenation of the following components: **Contextualized Embedding**, We feed the input document D, i.e., $[CLS]w_1w_2 \ldots w_n[SEP]$ to the pre-trained BERT_{base} transformer and take the hidden states of the last layer of the BERT model, i.e., $E = [e_1, e_2, \ldots, e_n]$, as the contextualized word embedding of the input document. Note that for the words that have multiple word-pieces, we take the average of their word-piece embeddings obtained from the BERT model. **Position Embedding** For each word w_i, we compute its distance to the subject w_s and the object w_o, i.e., $d^i_s = \|i - s\|$ and $d^i_o = \|i - o\|$, respectively. The distances are represented using high dimensional vectors e^s_i and e^o_j obtained from randomly initialized embedding tables. During training, the embedding tables are being updated. **Entity Type Embedding** The type of the two entities (i.e., Quantity, Measured-Entity, Measured-Property, and Qualifier) are represented using high dimensional vectors obtained from randomly initialized embedding tables. The embedding tables will be fine-tuned during training.

The concatenation of the aforementioned embedding vectors, i.e., $X = [x_1, x_2, \ldots, x_n]$, are used

to represent the words of the input document. It is noteworthy that since the parameters of the pre-trained BERT$_{base}$ are fixed during training, in order to tailor the contextualization of the word embeddings to this task, we feed the vectors X to a Bi-directional Long Short-Term Memory (BiLSTM) network and we use the hidden states of the BiLSTM neurons, i.e., $H = [h_1, h_2, \ldots, h_n]$, as the final vector representations of the input document D. The vectors H will be used by the subsequent components.

2.2 Dependency Path Reasoning

To find the dependency path between the subject and the object entities, we employ a translation-based perspective. More specifically, given the vector representations of the subject entity, i.e., h_s, and the object entity, i.e., h_o, the dependency path should be represented using the vector P such that using this vector, the subject representation h_s is transferred (i.e., translated) to the object representation h_o, under the operation Φ. Formally, $h_o = \Phi(h_s, P)$. Using this definition, we can define the path representation by P by exploiting the inverse operation Φ^{-1}, i.e., $P = \Phi^{-1}(h_s, h_o)$. After obtaining the path representation P, we compare it with the representations of the other words of the document D to assess their likelihood to be included in the dependency path. Concretely, the similarity between the vector h_i and the vector P could be used to estimate the probability of the word i to be used in the dependency path. However, one limitation of this method is that the likelihood of the word w_i is computed regardless of the other words w_j where $j \notin \{i, s, o\}$. To address this issue, we propose to compute the likelihood of the word w_i based on the interaction between the representation of the word w_i, i.e., h_i, the representations of the other words, i.e., h_j for $j \notin \{i, s, o\}$, and the path representation P. To this end, we first compute a vector representation for the words w_j by applying MAX_POOL operation on all words w_j for $j \notin \{i, s, o\}$: $\bar{h}_{-i} = MAX\_POOL(h_1, h_2, \ldots, h_j)$. Afterwards, we apply the function Φ^{-1} on the vectors P and $\bar{h}_{-i}$: $\hat{h}_i = \Phi^{-1}(\bar{h}_{-i}, P)$. The vector $\hat{h}_i$ represents the path for transferring (i.e., translating) the vector $\bar{h}_{-i}$ to P. As such, the similarity between $\hat{h}_i$ and h_i could reveal how important is the word w_i to convert the representation of the context w_j for $j \notin \{i, s, o\}$ to the representation of the dependency path P. Therefore, we use this similarity, i.e., $Sim_i = \left\| \hat{h}_i - h_i \right\|$, as the score of the word w_i to be included in the dependency path. The words that their score is above a pre-defined threshold will be used as the inferred dependency path.

It is worth noting that to learn the function Φ^{-1}, in this work, we use a feed forward neural network. In particular, the concatenation of the vectors h_s and h_o are fed into a 2-layer feed forward neural network with $|P|$ neurons at the final layer: $P = FF([h_s : h_o])$, where [:] represents concatenation and FF represents the feed-forward neural network. To train the FF network for the RE task, we use the vector P to predict the probability distribution $P_\Phi(\cdot|D, t, a)$ using another feed-forward network FF_2 whose final layer dimension equals the number of labels, i.e., $|L|$. We use negative log-likelihood to train the FF and FF_2 networks: $\mathcal{L}_\Phi = -log(P_\Phi(l|D, t, a))$ where l is the gold label.

Finally, to represent the induced path, we take the max-pooled representation of the words in the path: $h_P = MAX\_POOL$ $(h_1, h_2, \ldots, h_p)$ where p is the number of words in the induced dependency path. The path representation h_p will be used by the subsequent components.

2.3 Regularization

Although the induced dependency path from the previous component is intended to contain the important information for the RE task, it might still contain some noisy information due to the contextualization in the input encoder. To overcome this noisy information, in this work, we propose to exploit the induced path as the information bottleneck (IB) (Tishby et al., 2000). IB's goal is to reduce the mutual information between the input and the bottleneck, meanwhile, to increase the mutual information between the bottleneck and the output. For the second goal, the bottleneck (i.e., the dependency path representation h_p) will be used by the prediction component, and the increase of its mutual information with the output is enforced by reducing the training loss (e.g., negative log-likelihood). To fulfill the first goal, i.e., decreasing the mutual information between the input and the bottleneck, we resort to a contrastive learning paradigm to estimate the mutual information between two high-dimensional vectors using the classification loss of a binary-discriminator. More specifically, the path representation h_p is concatenated with the

max-pooled representation of the input document D, i.e., $h_d = MAX\_POOL(h_1, h_2, \ldots, h_n)$, and this concatenation, i.e., $h_{pos} = [h_p : h_d]$, serves as the positive sample for the contrastive learning. To construct the negative samples, we first take the max-pooled representation of a randomly chosen document D' from the same mini-batch, i.e., $h_{d'} = MAX\_POOL(h'_1, h'_2, \ldots, h'_m)$ where h'_i is the representation of the i-th word in the document D' and m is the total number of words in D'. Afterwards, the concatenation of h_p and $h_{d'}$ is employed as the negative sample: $h_{neg} = [h_p : h_{d'}]$. Finally, a feed-forward discriminator is employed and trained to distinguish the positive samples from the negative ones, i.e., $\mathcal{L}_{disc} = \log(1 + e^{(1-D(h_{pos}))}) + \log(1 + e^{D(h_{neg})})$. By adding the discriminator loss $\mathcal{L}_{disc}$ to the final loss function and decreasing it, the estimated mutual information between the input and the bottleneck (i.e., the path representation h_p) is decreased too.

2.4 Prediction

To make the final prediction on the relation between the given subject and object entities, we employ the representations of the induced dependency path (i.e., h_p), the subject entity (i.e., h_s), and the object entity (i.e., h_o) to construct the final vector $V = [h_p : h_s : h_o]$ where [:] represent concatenation. The vector V is finally consumed by a feed-forward neural network to predict the distribution $P(\cdot|D, t, a)$. The loss function to train the main RE task is thus defined as: $\mathcal{L}_{pred} = -log(P(l|D, t, a))$ where l is the gold label. The overall loss function to train the entire model is: $\mathcal{L} = \mathcal{L}_{pred} + \alpha\mathcal{L}_{\Phi} + \beta\mathcal{L}_{disc}$ where α and β are the trade-off parameters.

3 Experiments

3.1 Dataset, Hyper-Parameters & Baselines

In order to demonstrate the effectiveness of the proposed model, i.e., Dynamic Path Reasoning (DPR), we evaluate it on the recent SemEVal 2021 Task 8 dataset. This dataset provides measurement annotation for 233 training documents, 65 development documents, and 130 testing documents, all in English. Note that we do experiments only on the train and trial set (as the gold entities are not available for test set). Also, we evaluate the model only for relation extraction, not the entire task (as such, we did not make a submission during MeasEval evaluation phase). More specifically, for each document, the positions of the measured entities, measured properties, quantities, and qualifiers are provided. Furthermore, for each measurement component, its relations with the other components or extra information (e.g., unit of quantity) is available. Note that in our experiments, we do not use the *annotation set* information which indicates which components belong to the same measurement.

We fine-tune the hyper-parameters of the proposed model on the development set of the SemEval 2021 Task 8 dataset. The model with the best performance on the development set is evaluated on the test set. Based on our experiments, the following hyper-parameters are selected: 50 dimensions for the position embedding and entity type embedding; 200 dimensions for the hidden layer of the BiLSTM and all feed-forward networks; 0.1 and 0.05 for the trade-off parameters α and β; 0.7 for the threshold in the dynamic path reasoning component; Adam optimizer with learning rate 0.3; batch-size 50; and early stopping with the patience of 10.

To comprehensively evaluate the proposed model, we compare its performance against the following baselines: (i) Sequential Models, specifically we compare with **BiLSTM** which takes the non-contextualized word embeddings of the input document (i.e., GloVe) and encode the sequence of the words. Moreover, we also compare with **BERT** model fine-tuned during training for the MRE task. (ii) Structure-aware models, these models employ the structure of the input document (e.g., dependency trees of the sentences). Specifically, we compare with iDepNN (Gupta et al., 2019) which employs the dependency trees of the sentences of the document. This baseline adds an edge between the roots of the trees to create a connected graph, Furthermore, it prunes the tree along the dependency path between the two entities of interest. Finally, we compare our model with **LSR** which dynamically infer a graph structure for the input document using the representations of the entities and other words on the dependency path between the entities.

3.2 Results

The results on the test set are presented in Table 1. There are several observations from this table. First, the proposed model significantly (with $p < 0.01$) outperforms the baselines. It indicates the importance of using dynamic path reasoning and also the

Model	Precision	Recall	F1
BiLSTM	65.3	71.1	68.1
BERT	70.4	71.8	71.1
iDepNN	69.4	75.0	72.4
LSR	72	75.9	73.9
DPR (Ours	70.1	83.4	**76.2**

Table 1: Performance on Test set

Model	Precision	Recall	F1
Full	73.2	86.7	79.4
Full^{-DPR}	71.1	79.1	74.9
Full^{DPRS}	70.2	82.3	75.8
Full^{-Reg}	72.9	80.6	76.6
Full^{dot}	73.8	76.4	75.1

Table 2: Performance of the ablated models on the development set

proposed regularization method. Second, Comparing the structure-aware and sequence-based baselines, it is evident that the structure of the input document is necessary for achieving better results. However, between the iDepNN and the LSR baseline, the latter has better performance due to its capability of inferring the structure of the document instead of relying on external parse trees as in iDepNN. Finally, this experiment shows that using the pre-trained language model BERT substantially improves the performance compared to a sequence-based model that utilizes GloVe embedding. This is on par with the recent advancement on NLP using contextualized word embeddings.

3.3 Ablation Study

In this section, we provide more insight into the effectiveness of different components of the proposed model. The major two components in our model are dynamic path reasoning and regularization. To study their importance, we evaluate the performance of the following baselines on the development set of the SemEVal 2021 Task 8 datasets: (i) **Full**$^{-DPR}$, this baseline completely removes the dynamic path reasoning component. More specifically, the vector h_p is removed from the final prediction vector V and the loss function $\mathcal{L}_\Phi$ is also removed from the overall loss function $\mathcal{L}$; (ii) **Full**^{DPRS}, this baseline employs the dynamic path reasoning component. However, to compute the similarity score Sim_i, instead of considering the context if the word w_i, it directly computes the score by $Sim_i = \|P - h_i\|$; (iii) **Full**$^{-Reg}$, this model complete remove the regularization component, i.e., by removing the loss function $\mathcal{L}_{disc}$ from the overall loss function $\mathcal{L}$; (iv) **Full**^{dot}, this ablated model preserves the regularization component. However, instead of using Information Bottleneck, it directly decreases the similarity between the path representation, i.e., h_p, and the input document representation, i.e., h_d, by replacing the $\mathcal{L}_{disc}$ by $\mathcal{L}_{\lceil \rfloor \sqcup} = h_p \cdot h_d$.

The results are presented in Table 2. This table shows that all components of the proposed model are necessary to achieve the highest performance. More specifically, the dynamic path reasoning has the highest impact on the performance as removing it will hurt the most. Also, it shows that the consideration of the context to compute the score for each word to be included in the induced path is necessary. Finally, it shows that regularization is helpful for exclude noisy information from the input. More interestingly, replacing the IB with a dot product to enforce the regularization hurts more than removing the regularization itself. It indicates the necessity of using IB for regularization.

4 Related Work

Measurement Relation Extraction (MRE) is one specific formulation of the general Relation Extraction (RE) task. In the literature, RE has been tackled by feature-based methods (Zelenko et al., 2003; Zhou et al., 2005; Sun et al., 2011; Nguyen and Grishman, 2014; Nguyen et al., 2015c) and advanced deep learning models (Zeng et al., 2014; Wang et al., 2016; Lee et al., 2017; Zhang et al., 2017; Nguyen et al., 2019; Jin et al., 2018; Veyseh et al., 2020b). Recently, structure-aware deep models have shown significant improvement for RE (Peng et al., 2017; Song et al., 2018; Xu et al., 2015; Liu et al., 2015; Miwa and Bansal, 2016; Nguyen and Grishman, 2018a; Zhang et al., 2018). For a thorough review of the prior works, refer to the recent work (Gupta et al., 2019; Nan et al., 2020; Veyseh et al., 2020a)

5 Conclusion

We proposed a new model for the MRE task. The introduced model employs a dynamic path reasoning component which induces important context words to predict the relation between two measurement components. Furthermore, we proposed a novel regularization method based on Information Bottleneck to exclude noisy information from the input. Our experiments on the SemEval 2021 Task 8 reveal the effectiveness of the proposed model.

Acknowledgments

This research has been supported by the Army Research Office (ARO) grant W911NF-21-1-0112. This research is also based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA), via IARPA Contract No. 2019-19051600006 under the Better Extraction from Text Towards Enhanced Retrieval (BETTER) Program. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of ARO, ODNI, IARPA, the Department of Defense, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein. This document does not contain technology or technical data controlled under either the U.S. International Traffic in Arms Regulations or the U.S. Export Administration Regulations.

References

Pankaj Gupta, Subburam Rajaram, Hinrich Schütze, and Thomas Runkler. 2019. Neural relation extraction within and across sentence boundaries. In *AAAI*.

Corey Harper, Jessica Cox, Curt Kohler, Antony Scerri, Ron Daniel Jr., and Paul Groth. 2021. SemEval 2021 task 8: MeasEval – extracting counts and measurements and their related contexts. In *Proceedings of the Fifteenth Workshop on Semantic Evaluation (SemEval-2021)*, Bangkok, Thailand (online). Association for Computational Linguistics.

Di Jin, Franck Dernoncourt, Elena Sergeeva, Matthew McDermott, and Geeticka Chauhan. 2018. MIT-MEDG at SemEval-2018 task 7: Semantic relation classification via convolution neural network. In *Proceedings of The 12th International Workshop on Semantic Evaluation*, pages 798–804.

Ji Young Lee, Franck Dernoncourt, and Peter Szolovits. 2017. MIT at SemEval-2017 task 10: Relation extraction with convolutional neural networks. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 978–984.

Yang Liu, Furu Wei, Sujian Li, Heng Ji, Ming Zhou, and Houfeng Wang. 2015. A dependency-based neural network for relation classification. In *ACL*.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using lstms on sequences and tree structures. In *ACL*.

Guoshun Nan, Zhijiang Guo, Ivan Sekulic, and Wei Lu. 2020. Reasoning with latent structure refinement for document-level relation extraction. In *ACL*.

Thien Huu Nguyen and Ralph Grishman. 2014. Employing word representations and regularization for domain adaptation of relation extraction. In *ACL*.

Thien Huu Nguyen and Ralph Grishman. 2018a. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*.

Thien Huu Nguyen, Barbara Plank, and Ralph Grishman. 2015c. Semantic representations for domain adaptation: A case study on the tree kernel-based method for relation extraction. In *ACL-IJCNLP*.

Tuan Ngo Nguyen, Franck Dernoncourt, and Thien Huu Nguyen. 2019. On the effectiveness of the pooling methods for biomedical relation extraction with deep learning. *arXiv preprint arXiv:1911.01055*.

Nanyun Peng, Hoifung Poon, Chris Quirk, Kristina Toutanova, and Wen-tau Yih. 2017. Cross-sentence n-ary relation extraction with graph lstms. In *TACL*.

Linfeng Song, Yue Zhang, Zhiguo Wang, and Daniel Gildea. 2018. N-ary relation extraction using graph state lstm. In *EMNLP*.

Ang Sun, Ralph Grishman, and Satoshi Sekine. 2011. Semi-supervised relation extraction with large-scale word clustering. In *ACL*.

Naftali Tishby, Fernando C Pereira, and William Bialek. 2000. The information bottleneck method. In *arXiv preprint physics/0004057*.

Amir Veyseh, Franck Dernoncourt, My Thai, Dejing Dou, and Thien Nguyen. 2020a. Multi-view consistency for relation extraction via mutual information and structure prediction. In *AAAI*.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Dejing Dou, and Thien Huu Nguyen. 2020b. Exploiting the syntax-model consistency for neural relation extraction. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8021–8032.

Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. 2016. Relation classification via multi-level attention cnns. In *EMNLP*.

Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. 2015. Classifying relations via long short term memory networks along shortest dependency paths. In *EMNLP*.

Dmitry Zelenko, Chinatsu Aone, and Anthony Richardella. 2003. Kernel methods for relation extraction. In *Journal of machine learning research*.

Daojian Zeng, Kang Liu, Siwei Lai, Guangyou Zhou, and Jun Zhao. 2014. Relation classification via convolutional deep neural network. In *COLING*.

Yuhao Zhang, Peng Qi, and Christopher D Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *EMNLP*.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D Manning. 2017. Position-aware attention and supervised data improve slot filling. In *EMNLP*.

Guodong Zhou, Jian Su, Jie Zhang, and Min Zhang. 2005. Exploring various knowledge in relation extraction. In *ACL*.