Improving First Order Temporal Fact Extraction with Unreliable Data

Bingfeng Luo[1], Yansong Feng[1*], Zheng Wang[2], and Dongyan Zhao[1]

[1] Institute of Computer Science and Technology, Peking University, P.R. China.,
bingfeng_luo, fengyansong, zhaody@pku.edu.cn
[2] School of Computing and Communications, Lancaster University, UK,
z.wang@lancaster.ac.uk

Abstract. In this paper, we deal with the task of extracting first order temporal facts from free text. This task is a subtask of relation extraction and it aims at extracting relations between entity and time. Currently, the field of relation extraction mainly focuses on extracting relations between entities. However, we observe that the multi-granular nature of time expressions can help us divide the dataset constructed by distant supervision into reliable and less reliable subsets, which can help to improve the extraction results on relations between entity and time. We accordingly contribute the first dataset focusing on the first order temporal fact extraction task using distant supervision. To fully utilize both the reliable and the less reliable data, we propose to use curriculum learning to rearrange the training procedure, label dropout to make the model be more conservative about less reliable data, and instance attention to help the model distinguish important instances from unimportant ones. Experiments show that these methods help the model outperform the model trained purely on the reliable dataset as well as the model trained on the dataset where all subsets are mixed together.

Keywords: temporal fact extraction, distant supervision, knowledge base

1 Introduction

Knowledge base population aims at automatically extracting facts about entities from text to extend knowledge bases. These facts are often organized as (*subject*, *relation*, *object*) triples. Among these relations, the relations that require time expressions as objects play an important role in the completeness of knowledge base. For example, every person should have *date_of_birth* and almost all asteroid should have *date_of_discovery*. However, we find that in Wikidata[3], 19.3% people do not have *date_of_birth*, and 39.3% asteroids do not have *date_of_discovery*. Therefore, extracting relations between entity and time is an important task.

[3] www.wikidata.org. It is a rapid-growing knowledge base and Freebase (www.freebase.com) is migrating its data to it.

As suggested by T-YAGO [20], which extends YAGO [6] with temporal aspects, there are two types of facts that we need to extract: *First order facts* are triples like ($Barack\_Obama$, spouse, $Michelle\_Obama$), whose subjects are entities. *Higher order facts* take other facts as subjects. For example, (($Barack\_Obama$, $spouse$, $Michelle\_Obama$), $start\_time$, $3\_October\_1992$) is a higher order fact indicating the start time of the marriage of Barack Obama and Michelle Obama. In this paper, we will focus on first order temporal fact extraction, which is a subtask of first order fact extraction (often referred to as *relation extraction*) that focuses on extracting facts that take time expressions as objects

Previous work of relation extraction mainly focuses on extracting relations between entities, and the task of extracting first order temporal facts receives only limited attention. Some researchers try to extract first order temporal facts from semi-structured documents like Wikipedia[4] [9], but how to extract first order temporal facts from free text is seldom investigated specifically.

Indeed, most of the existing methods developed to extract relations between entities can be used directly to extract first order temporal facts, which to some extent explains the sparsity of researches in first order temporal fact extraction. However, there are still interesting properties in first order temporal fact extraction that the entity-entity relation extraction task does not have.

When extracting first order facts from free text, distant supervision is often used to build noisy datasets [12]. Given a (subject s, relation r, object o) triple in a knowledge base, it uses s and o to retrieve text corpora like Wikipedia articles, and collects sentences containing both s and o as supports for this triple. The noisy nature of distant supervision has long been bothering researchers, and various techniques have been introduced to deal with the noise [17, 16].

However, it is different when applying distant supervision to first order temporal fact extraction. We find that the more fine-grained the time expression is, the more likely the retrieved sentence is a true support for this triple. For example, sentences containing both *Oct. 5, 2011* and *Steve Jobs* are highly likely to indicate Steve Jobs' death date, while sentences containing only *2011* and *Steve Jobs* may only talk about his resignation. The intuition is that, there is usually only one important thing that relates to an entity in a single day. As the time granularity becomes coarser, more important things are likely to happen in the same time period, and hence the data quality goes down.

We find that sentences containing full date (day, month and year) are highly reliable and we can train a relation extractor as if we are using human labeled data. However, there is still useful knowledge remaining in sentences containing only coarser granularities. Therefore, how to use the less reliable data to improve the model becomes another problem.

Following this observation, we construct the first first order temporal fact extraction dataset with distant supervision. The dataset is grouped into 4 smaller dataset with decreasing reliability. To fully utilize both the reliable and less reliable data, we propose to use curriculum learning to rearrange the training procedure, label dropout to make the model more conservative about less reliable

[4] www.wikipedia.org

data, and instance attention to help the model distinguish important instances from unimportant ones. Experiments show that these methods help the model outperform the model trained purely on the reliable dataset as well as the model trained on the dataset where all subsets are mixed together.

2 Related Work

Relation Extraction The most related thread of work to us is relation extraction. Most of the methods in this field can be applied directly to first order temporal fact extraction, which possibly explains why there are seldom researches that focus specifically on extracting first order temporal facts from free text. The commonly used paradigm in relation extraction is distant supervision [12], which tries to construct a noisy dataset using triples in a knowledge base as guidance. Feature based [12], graphic model based [7, 16] and neural network based [23] methods have been applied under this paradigm. To cope with the noisy nature of distant supervision, there are also some researches aim at reducing the noise introduced by distant supervision [17], or put this task in the multi-instance paradigm [7, 16, 23]. In first order temporal facts extraction, we find that the multi-granular nature of time expressions can help us distinguish reliable distant supervision data from less reliable ones, and we can use this property to improve the model performance.

Higher Order Temporal Fact Extraction Higher order temporal fact extraction mainly aims at identifying the valid time scope of a (subject, relation, object) triple. Therefore, it is also referred to as temporal scoping. The commonly used dataset is introduced by the 2011 and 2013 temporal slot filling (TSF) shared tasks hosted by Text Analysis Conference (TAC) [8, 5]. Regular expression based methods[20], graph based methods [19] and distant supervision based methods [2, 15] have been used in this task. While our task focuses on first order temporal facts which contain a variety of relations (see Table 1), this thread of work only tries to find the start time and end time of a triple, which makes this task seems easier. However, since this task takes a triple as subject, people need to come up with different methods to handle this property, and thus makes this task harder.

Event Temporal Relation Identification Event temporal relation identification is a related task introduced by TempEval [18, 14]. This task aims at identifying event-time and event-event temporal relations like $before$, $after$ and $overlap$. Feature based methods [11] and Markov Logic Network based methods [22] have been applied to this task. Apart from ordering events, TIE system [10] also uses probabilistic inference to bound the start and the ending time of events. This task differs from our task in that it deals with the relations between event and time rather than entity and time, and it mainly focuses on ordering events while we focuses on extracting triples that take time expressions as objects.

3 Dataset Construction

We use Wikidata as the knowledge base and Wikipedia articles as the corpus. There are about 29 relations that require time as object in Wikidata. To ensure sufficient training instances for each relation, 12 relations are used (see Table 1).

Distant Supervision For each (entity e, relation r, time t) triple, we use the official name and the aliases in Wikidata of entity e as its surface forms. As for time t, we generate its surface forms in 4 granularities: *Full Date* (like 12 Jan. 2011), *Month Year* (like Jan. 2011), *Year Only* (like 2011), and *Month Day* (like 12 Jan.).

Negative Data The negative data come from two sources. First, for every entity mention $\tilde{e}$ in a retrieved sentence, each $(\tilde{e}, t)$ pair except (e, t) is considered to have no relation. Entity mentions are identified by finding strings that matches the canonical names or aliases of entities in Wikidata using Aho-Coraisake algorithm [1]. Second, we retrieve all the sentences containing the surface forms of entity e and detect time mentions $\tilde{t}$ with SUTime [4]. For each entity surface form, we find at most 5 corresponding entities using Wikidata API[5]. If $\tilde{t}$ does not appear in any triples that take entity e as subject in Wikidata, the entity mention and the time mention are considered to have no relation.

Dataset Validation We manually examined 20 randomly selected sentences for each relation in each granularity (see Table 1). We find that full-date data have high quality and the data quality goes down as the granularity becomes coarser. Since sentences containing only day and month are limited, some relations do not have 20 instances in month-day data. Considering the limited number and low quality, month-day data will not be used in the experiment. Note that *point_in_time* and *end_time* data in full-date granularity are not very reliable. This is because that these two relations often take battles as subjects. However, descriptions about battles are often very detailed, and many retrieved sentences actually describe specific events in the battle rather than the battle itself.

4 Model

The inputs to our first order temporal fact extraction model are sentences labeled with an entity mention e and a time mention t. The task is to identify the relation between the given time and entity. We will first briefly introduce the baseline model. Then we will discuss several methods to utilize both the reliable and the unreliable data to achieve better performance.

[5] www.wikidata.org/w/api.php

Relation	Full Date	Month Year	Year Only	Month Day
date_of_birth	**20/20**	**19/20**	**19/20**	11/20
date_of_death	**20/20**	13/20	13/20	9/20
time_of_discovery	**20/20**	15/20	12/20	5/20
inception	**17/20**	15/20	7/20	4/20
dissolved_or_abolished	15/20	8/20	10/20	1/5
time_of_spacecraft_launch	**19/20**	**17/20**	13/20	1/1
time_of_spacecraft_landing	**18/20**	5/20	7/20	4/5
first_performance	**18/20**	**16/20**	15/20	1/2
publication_date	**19/20**	15/20	**16/20**	5/12
point_in_time	13/20	13/20	**19/20**	8/19
start_time	**18/20**	**18/20**	14/20	14/18
end_time	10/20	13/20	9/20	11/20

Table 1. Statistics of extracted temporal relation mention. Left side of the slash is the number of correct relation mentions, right side is the number of mentions examined.

4.1 PCNN Model

Our basic model is the Piecewise Convolutional Neural Network (PCNN) [23], which achieves the state-of-art results in entity-entity relation extraction task. Following the PCNN work, we also concatenate position embeddings to the original word embedding as input. To be concrete, for each word in the sentence, we calculate its distance to the entity mention and the time mention. Each distance (e.g. -2, -1, 1, 2, etc) is associated with a randomly initialized embedding vector, which will be updated during training.

As shown in Figure 1, the input sentence is divided into three parts by the entity mention e and the time mention t. The convolution and max-pooling operation are applied to the three parts separately to obtain the embeddings of each part. After that, these three embeddings are concatenated and fed to a full connection layer. Finally, the softmax classifier is used to generate the relation distribution, and cross entropy is used as loss function.

Due to the noise in the less reliable data, we find that the model trained on the dataset where all the subsets (except the month-day data) mixed together performs significantly worse than the model trained only on the full-date data. Therefore, we consider the latter one as our baseline model. Due to the high reliability of full-date data, we use multi-class classification paradigm rather than the multi-instance classification paradigm used in the original PCNN paper.

4.2 Curriculum Learning

Instead of adding less reliable data directly to the training set, an alternative way to incorporate these data is to use the idea of curriculum learning [3]: start with easier aspect of the task and then increase the difficulty level gradually.

To be concrete, we first train our model on the reliable full-date data for c_1 epochs. Since the data are highly reliable, the task is relatively easy for the

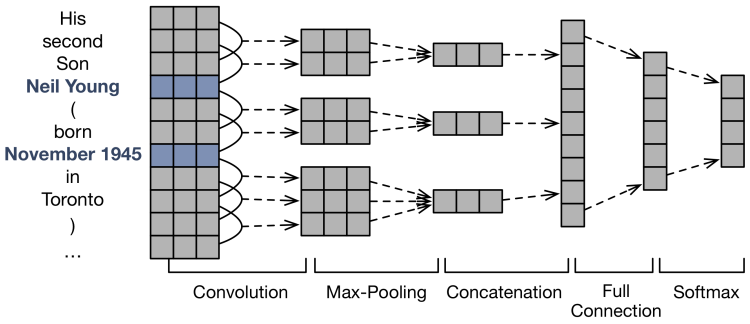


Fig. 1. The architecture of the PCNN model. In this example, *Neil Young* is the entity mention and *November 1945* is the time mention.

model and can give the model the basic classification ability. After that, we add the less reliable month-year data to the training set to increase the difficulty and train for another c_2 epochs. Finally, the most unreliable year-only data are added, and the model is trained for the last c_3 epochs.

The intuition behind this is that, after training on the reliable data for several epochs, the model has already been positioned near the optimal point in the space of model parameters, and therefore it is less likely to be misled by less reliable data. Also note that since the reliable data are used throughout the entire training procedure and is fitted more times than less reliable data, the reliable data actually lead the entire training procedure.

4.3 Label Dropout

Inspired by the DisturbLabel method [21], which randomly convert the gold label of a training instance to another label arbitrarily, we propose a more conservative method called Label Dropout to exploit less reliable data.

While DisturbLabel adds noise to the dataset to make the model more robust and generalize better, we want to use noise to make the model more conservative. Rather than converting the gold label of a training instance to another label arbitrarily, we only convert it to the negative label. To be concrete, for each positive training instance, we randomly convert it to negative instance with probability p in each epoch, where p decreases with data reliability.

The intuition is that, if a positive instance turns negative randomly during training, it actually has two opposite impact on the model parameters, which means it will have less influence on the model and the model will be more conservative about this unreliable instance and gives it lower score. Apart from that, with the help of the introduced noise, this method also forces the model to learn the most essential features in less reliable data, and thus avoids the model from learning unreliable features.

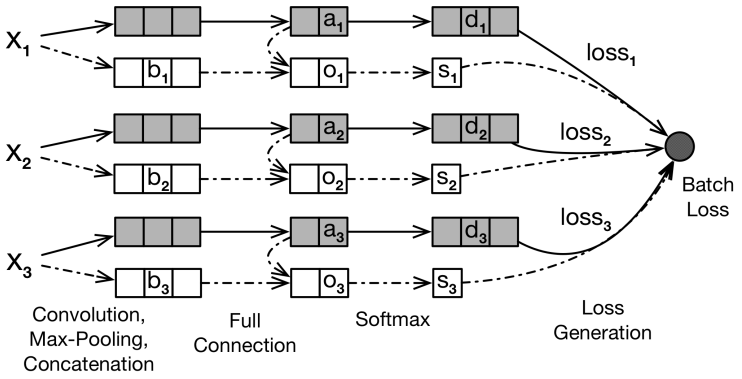


Fig. 2. Instance Attention Model. The part with solid squares and solid lines is the vanilla PCNN for relation extraction. The part with hollow squares and dashed lines is the W-PCNN used to obtain weights for each training instance $\mathbf{x_i}$. $\mathbf{a_i}$ is the output of the full connection layer of PCNN, $\mathbf{b_i}$ is the output of the connection layer of W-PCNN, $\mathbf{o_i}$ is the output of the full connection layer of W-PCNN, s_i is the instance weight and $\mathbf{d_i}$ is the relation distribution.

4.4 Instance Attention

To make the training procedure more robust, we further introduce a novel instance attention method. The intuition is that we want the model to be able to distinguish important instances from unimportant ones. This is achieved by applying attention mechanism to the instance level. In each batch, we generate a weight for each instance to indicate its importance. On the one hand, under the framework of curriculum learning, the model can learn the pattern of noisy data and give them lower weights. On the other hand, the model can learn to give up very hard instances or concentrate on promising instances that is not well-learned.

To be more specific, we train a parallel PCNN model containing both the original PCNN and a smaller PCNN (referred to as W-PCNN) with fewer convolution kernels designed to produce the weight for each instance $\mathbf{x_i}$. As shown in Figure 2, the part with solid squares and solid lines is the original PCNN model, and the part with hollow squares and dashed lines is the W-PCNN. To consider both the information of the input instance and the PCNN prediction, we concatenate the output vector $\mathbf{a_i}$ of the full connection layer in the original PCNN and the output vector $\mathbf{b_i}$ of the concatenation layer in W-PCNN. The concatenated vector $\mathbf{c_i} = [\mathbf{a_i}^T, \mathbf{b_i}^T]^T$ is fed to a full connection layer, generating an output vector $\mathbf{o_i}$:

$$\mathbf{o_i} = \mathbf{W_o} \times \mathbf{c_i} \tag{1}$$

where $\mathbf{W_o}$ is the weight matrix. The weight of the input instance $\mathbf{x_i}$ is generated by instance level softmax:

$$s_i = \frac{e^{\mathbf{w_s}^T \mathbf{o_i}}}{\sum_{i=1}^{k} e^{\mathbf{w_s}^T \mathbf{o_i}}} \tag{2}$$

where $\mathbf{w_s}$ is the weight vector, k is the batch size and the denominator is the used for normalization. The final batch loss is then generated by:

$$batch\_loss = \sum_{i=1}^{k} s_i \times loss_i \tag{3}$$

where $loss_i$ is the cross entropy loss of instance $\mathbf{x_i}$ generated by the original PCNN model .

5 Experiments

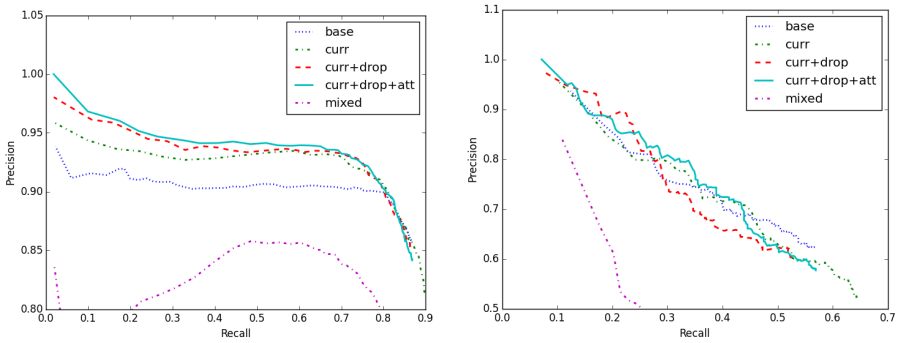
Dataset Detail We use the full-date data as our basic dataset. With 8:1:1 split, we get 22,214 positive instances for training, 2,776 for validation and 2,771 for testing.[6]. Since *date_of_birth* and *date_of_death* (*big relations*) take a large portion of the data, we only use relations other than them in month-year and year-only data as additional unreliable data, which contains 2,094 and 53,469 positive instances separately. We generate 2 negative instances for every positive instance using the first strategy. As for the second strategy, we generate 40,000 negative instances for training, 3,576 for validation, and 3,493 for testing.

Hyperparameters We use 100-dimension word embedding pre-trained using GloVe [13] and 20 dimensional randomly initialized position embedding. We use SGD for optimization with batch size 20, learning rate 0.1, dropout probability 0.5. The PCNN model has 200 convolution kernels followed by a full connection layer with 200 output units. W-PCNN has 50 convolution kernels and the full connection layer has 200 output units. The settings of curriculum learning parameters are $c_1 = c_2 = c_3 = 15$, which means the month-year data are added in the 16^{th} epoch, and the year-only data are added in the 31^{th} epoch. The label dropout method drops month-year data with probability 0.5, and year-only data with probability 0.7 (we do not conduct label dropout for full-date data).

5.1 Main Results

Following previous work on relation extraction, we report the precision recall curve (PR curve) of our model. The overall PR curve on test set is shown in Figure 3(a). We can see that if we do not distinguish the subsets with different reliability and mix them together (see the *mixed* line), we will get significant worse results than the model trained with only reliable subset. Therefore, we

[6] The dataset can be downloaded from: github.com/pfllo/TemporalFactExtraction



(a) Overall PR Curves (b) Small Relation PR Curves

Fig. 3. Precision recall curves on test data. The label *base* refers to the basic PCNN model trained only on the full-date data, *curr* refers to curriculum learning, *drop* refers to label dropout, *att* refers to instance attention, *mixed* refers to the model trained with these three types of data mixed together. The figure is truncated to better display the main results. Therefore, some parts of the *mixed* line are invisible.

consider the latter model as our baseline. After adding curriculum learning, the model gains a significant boost over the baseline PCNN. Label dropout improves the performance in the low recall region (corresponding to high precision). Instance attention further improves the results in both high and low recall region. This shows that all of the three proposed methods help the model make better use of the less unreliable data.

To further investigate where these improvements come from, we also report PR curves of *small relations* (relations other than *date_of_birth* and *date_of_death*) in Figure 3(b). As we can see, the small-relation curve of curriculum learning is close to the baseline, showing that the additional noisy instances does not contribute much useful information to small relations and the overall boost comes mainly from the improvement of big relations. Recall that we do not use big relations in less reliable data, therefore the reason for the improved performance of big relations should come mainly from the reduced false positive rate.

When label dropout is added, the noisy positive data become useful and make the model be more conservative about its prediction. When the model assigns a positive label to an instance that it is not very confident about, it will produce a lower score than before, and thus makes the scores of confident predictions and less confident predictions more separable from each other, which is reflected by the better performance of low recall region. However, the lowered scores of less confident predictions make them less separable from those instances that the model is very unconfident about. Therefore, the performance decreases in the high recall region.

Finally, instance attention enables the model to avoid the influence of noise and very hard instances by giving them lower weights and concentrate on promising instances that are not well learned. This mechanism makes the training pro-

cedure more self-adaptive and thus smoothes the small relation curve of label dropout, which leads to good performance in both high and low recall regions.

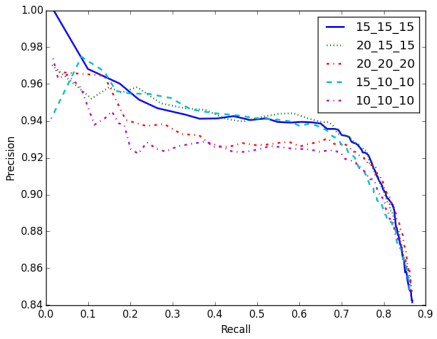


Fig. 4. Influence of curriculum learning parameters. The legend is c_1_c_2_c_3, corresponding to the epochs trained with only full-date data, the epochs trained with full-date and month-year data, and the epochs trained with all data.

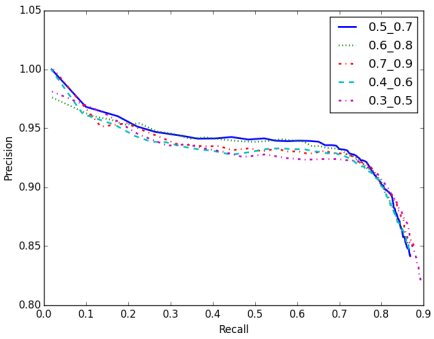


Fig. 5. Influence of label dropout parameters. The legend is in the format of p_1_p_2, which corresponds to the label dropout rate of month-year data and the label dropout rate of the year-only data respectively.

5.2 Influence of Curriculum Learning Parameters

Since the curriculum learning method contributes the major boost, it is worth further investigating how its parameters influence the result. We report the PR curves of our full model with different settings of c_i. As we can see from Figure 4, $c_1 = c_2 = c_3 = 15$ achieves the best overall performance. Slightly increasing c_1 or slightly decreasing c_2 and c_3 does not significantly influence the overall performance, which shows the robustness of the model.

However, when we decrease the epochs trained only with full-date data (c_2 and c_3 are also decreased to reduce their influence), the performance significantly drops. This indicates that we should train the model with only reliable data for enough epochs so that it is tuned near the optimal position in parameter space before exposed to less reliable data. Otherwise, the model will be easily misled to suboptimal positions by the noise. We can also see that increasing the epochs for less reliable data impairs the overall performance as well. This indicates that although our model has some resistance to the noise in less reliable data, it still tend to overfit the less reliable data given enough training epochs.

To conclude, when tuning the model, it is safer to use larger c for reliable data and smaller c for less reliable data. Note that 15 is the number of epochs that the baseline model needs to roughly converge, which to some extent explains why the model underfits the reliable data when c_1 is smaller than 15, and why the model overfits the less reliable data when c_2 and c_3 is bigger than 15.

5.3 Influence of Label Dropout Parameters

To see how the label dropout parameters influence the model performance, we report the PR curves of our full model with different settings of label dropout parameters in Figure 5. As we can see, the best performance is achieved when $p_1 = 0.5$ (label dropout rate of month-year data) and $p_2 = 0.7$ (label dropout rate of year-only data). Slightly increase the label dropout rate does not produce significant difference, which to some extent shows the model robustness. Further increasing the label dropout rate will produce the results that are consistently worse, indicating that high label dropout rate will decrease the amount the information that the model can learn from the less reliable data. However, the performance drops immediately when the label dropout rate decreases. This indicates that the increased influence of unreliable positive data causes more harm to the model than the loss of learnable information.

6 Conclusion

In this paper, we contribute the first dataset that focuses specifically on first order temporal fact extraction using distant supervision. We observe that, by grouping the data with different time granularities, we can naturally obtain data groups with different levels of reliability. Although we can train our model directly on the reliable full-date data, methods like curriculum learning, label dropout and instance attention can further exploit the less reliable data and produce better results than the model trained with only reliable ones.

Acknowledgement

This work was supported by National High Technology R&D Program of China (Grant No. 2015AA015403, 2014AA015102), Natural Science Foundation of China (Grant No. 61202233, 61272344, 61370055) and the joint project with IBM Research. Any correspondence please refer to Yansong Feng.

References

1. Aho, A.V., Corasick, M.J.: Efficient string matching: an aid to bibliographic search. Communications of the ACM 18(6), 333–340 (1975)
2. Artiles, J., Li, Q., Cassidy, T., Tamang, S., Ji, H.: Cuny blender tackbp2011 temporal slot filling system description. In: Proceedings of Text Analysis Conference (TAC) (2011)
3. Bengio, Y., Louradour, J., Collobert, R., Weston, J.: Curriculum learning. In: Proceedings of the 26th annual international conference on machine learning. pp. 41–48. ACM (2009)
4. Chang, A.X., Manning, C.D.: Sutime: A library for recognizing and normalizing time expressions. In: LREC. pp. 3735–3740 (2012)
5. Dang, H.T., Surdeanu, M.: Task description for knowledge-base population at tac 2013 (2013)

6. Fabian, M., Gjergji, K., Gerhard, W.: Yago: A core of semantic knowledge unifying wordnet and wikipedia. In: 16th International World Wide Web Conference, WWW. pp. 697–706 (2007)
7. Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., Weld, D.S.: Knowledge-based weak supervision for information extraction of overlapping relations. In: Proceedings of ACL (2011)
8. Ji, H., Grishman, R., Dang, H.: Overview of the tac2011 knowledge base population track. In: Text Analysis Conference (2011)
9. Kuzey, E., Weikum, G.: Extraction of temporal facts and events from wikipedia. In: Proceedings of the 2nd Temporal Web Analytics Workshop. pp. 25–32. ACM (2012)
10. Ling, X., Weld, D.S.: Temporal information extraction. In: AAAI. vol. 10, pp. 1385–1390 (2010)
11. Mani, I., Verhagen, M., Wellner, B., Lee, C.M., Pustejovsky, J.: Machine learning of temporal relations. In: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics (2006)
12. Mintz, M., Bills, S., Snow, R., Jurafsky, D.: Distant supervision for relation extraction without labeled data. In: Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (2009)
13. Pennington, J., Socher, R., Manning, C.D.: Glove: Global vectors for word representation. In: EMNLP. vol. 14, pp. 1532–1543 (2014)
14. Pustejovsky, J., Verhagen, M.: Semeval-2010 task 13: evaluating events, time expressions, and temporal relations (tempeval-2). In: Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions. pp. 112–116. Association for Computational Linguistics (2009)
15. Sil, A., Cucerzan, S.: Temporal scoping of relational facts based on wikipedia data. CoNLL-2014 p. 109 (2014)
16. Surdeanu, M., Tibshirani, J., Nallapati, R., Manning, C.D.: Multi-instance multi-label learning for relation extraction. In: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning. pp. 455–465. Association for Computational Linguistics (2012)
17. Takamatsu, S., Sato, I., Nakagawa, H.: Reducing wrong labels in distant supervision for relation extraction. In: Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (2012)
18. Verhagen, M., Gaizauskas, R., Schilder, F., Hepple, M., Katz, G., Pustejovsky, J.: Semeval-2007 task 15: Tempeval temporal relation identification. In: Proceedings of the 4th International Workshop on Semantic Evaluations. pp. 75–80. Association for Computational Linguistics (2007)
19. Wang, Y., Yang, B., Qu, L., Spaniol, M., Weikum, G.: Harvesting facts from textual web sources by constrained label propagation. In: Proceedings of the 20th ACM international conference on Information and knowledge management (2011)
20. Wang, Y., Zhu, M., Qu, L., Spaniol, M., Weikum, G.: Timely yago: harvesting, querying, and visualizing temporal knowledge from wikipedia. In: Proceedings of the 13th International Conference on Extending Database Technology (2010)
21. Xie, L., Wang, J., Wei, Z., Wang, M., Tian, Q.: Disturblabel: Regularizing cnn on the loss layer. arXiv preprint arXiv:1605.00055 (2016)
22. Yoshikawa, K., Riedel, S., Asahara, M., Matsumoto, Y.: Jointly identifying temporal relations with markov logic. In: Proceedings of ACL-IJCNLP (2009)
23. Zeng, D., Liu, K., Chen, Y., Zhao, J.: Distant supervision for relation extraction via piecewise convolutional neural networks. EMNLP (2015)