Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Improving Event Causality Recognition with Multiple Background Knowledge Sources Using Multi-Column Convolutional Neural Networks

Canasai Kruengkrai, Kentaro Torisawa, Chikara Hashimoto, Julien Kloetzer, Jong-Hoon Oh, Masahiro Tanaka
National Institute of Information and Communications Technology, Kyoto, 619-0289, Japan
{canasai, torisawa, ch, julien, rovellia, mtnk}@nict.go.jp

Abstract

We propose a method for recognizing such event causalities as "smoke cigarettes" → "die of lung cancer" using background knowledge taken from web texts as well as original sentences from which candidates for the causalities were extracted. We retrieve texts related to our event causality candidates from four billion web pages by three distinct methods, including a why-question answering system, and feed them to our *multi-column* convolutional neural networks. This allows us to identify the useful background knowledge scattered in web texts and effectively exploit the identified knowledge to recognize event causalities. We empirically show that the combination of our neural network architecture and background knowledge significantly improves average precision, while the previous state-of-the-art method gains just a small benefit from such background knowledge.

1 Introduction

Event causality, such as "smoke cigarettes" → "die of lung cancer," is critical knowledge for many NLP applications, including machine reading and comprehension (Richardson, Burges, and Renshaw 2013; Berant et al. 2014), process extraction (Scaria et al. 2013), and future event/scenario prediction (Radinsky, Davidovich, and Markovitch 2012; Hashimoto et al. 2014). However, the state-of-the-art methods for event causality recognition still suffer from low precision and coverage because event causality is expressed in a wide range of forms that often lack explicit clues indicating the existence of event causality. Consider the following sentences:

1. Typhoons have strengthened *because* global warming has worsened.
2. Global warming worsened, and typhoons strengthened.

The first sentence includes "because," which explicitly indicates the event causality between effect "typhoons strengthen" and cause "global warming worsens." On the other hand, the second sentence has no such clues. Nonetheless, many people would infer that this sentence expresses the same event causality as that in the first sentence. This is possible because people have *background knowledge* about "typhoons" and "global warming."

keizaikankyou-ga akkasuru → *shijoukinri-ga gerakusuru* ("economic environment deteriorates" → "markets decline")
bukka-wa gerakusuru → *defure-ni naru* ("prices of commodities decline" → "become deflation")
kubi-ni naru → *sitsugyouhoken-o morau* ("get fired" → "get unemployment insurance")
chiyuryoku-o takameru → *atopii-o kokufukusu* ("enhance healing power" → "defeat atopic syndrome")
bitamin-ga fusokusuru → *koukakuen-ni naru* ("lack vitamin" → "cause angular cheilitis")

Table 1: Japanese examples of event causalities successfully recognized by our proposed method

This work develops a method that recognizes event causalities from sentences *regardless* whether they have such explicit clues as "because." The novelty of this work lies in that we exploit a wide range of *background knowledge* (written in web texts) using convolutional neural networks. In other words, given an event causality candidate, our neural network analyzes descriptions in web texts that are somehow *related* to the given causality candidate and judges whether the candidate is a proper causality. The web texts are retrieved by three distinct methods from our 4-billion-page web archive. We target such event causalities as "global warming worsens" → "typhoons strengthen," in which each cause phrase ("global warming worsens") and effect phrase ("typhoons strengthen") consists of a noun ("global warming") and a verb ("worsens"). Our experimental results showed that our neural network-based method outperforms the state-of-the-art method based on SVMs (Hashimoto et al. 2014) and its variants augmented with the background knowledge sources introduced in this work. This suggests that our neural network architecture is more suitable for dealing with background knowledge. Table 1 shows examples of event causalities successfully recognized by our method but not by Hashimoto et al.'s method.

As one method to extract background knowledge from web archives, we use a why-question answering (why-QA) system following (Oh et al. 2013) that retrieves seven-sentence passages as answers to a given why-type ques-

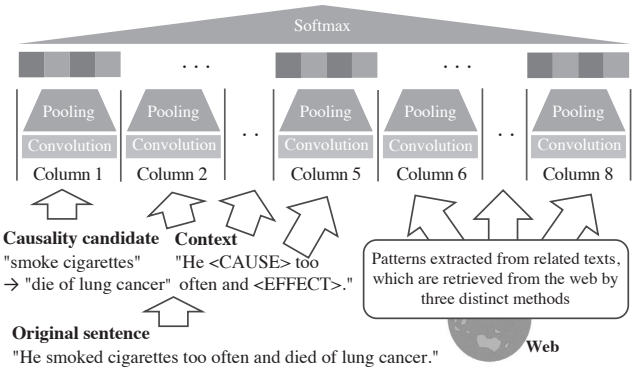

Figure 1: Our MCNN architecture

tion. We automatically generate a question from the effect part of an event causality candidate and extract background knowledge from its answers. For recognizing the event causality candidate, "global warming worsens" → "typhoons strengthen" as a proper causality, our method generates the question, "Why do *typhoons strengthen*?", and retrieves such answers as one that includes the following two sentences: "*Typhoons* are becoming more powerful than before. This is probably an effect of recent *global warming*." Useful background knowledge here might be the causal relation between "Typhoons are becoming more powerful" and "global warming," which somehow *resembles* the event causality candidate. A problem is that the why-QA system's answers have a wide variety of forms, and the recognition of such relations and their similarity to our target causality candidates is not a trivial task. Furthermore, not all of the answers necessarily express any useful background knowledge. Even proper answers to a given question may not provide any useful knowledge (e.g., "The paper reported that *typhoons* are becoming powerful due to high sea temperature. Another paper addressed the issues caused by *global warming*."). Extracting useful knowledge from such noisy texts itself is challenging.

To perform this difficult task, we use multi-column convolutional neural networks (MCNNs) (Ciresan, Meier, and Schmidhuber 2012), which are a variant of convolutional neural networks (Lecun et al. 1998) with several independent columns. Each column has its own convolutional and pooling layers, and the outputs of all the columns are combined in the last layer to provide a final prediction (Fig. 1). In our architecture, we use five columns to process the event causality candidates and their surrounding contexts in the original sentence and three other columns to deal with web texts retrieved by different methods.

Our work was inspired by the research of Hashimoto et al. (2014). They improved the performance of event causality recognition using a carefully selected set of *short* binary patterns that connect pairs of nouns, like "A causes B" and "A prevents B". For instance, to judge whether causality "smoke cigarettes" → "die of lung cancer" is proper, they checked whether "cigarettes" and "lung cancer" fill any of the binary patterns in a web archive. If such patterns exist, they are encoded in the features of their SVM classifier. Note that the set of short binary patterns they prepared includes not only CAUSE ("A causes B") relations but also such relations as MATERIAL ("A is made of B") and USE ("A is used for B"), which are not directly related to causality. They showed that such patterns actually improved the performance, suggesting that a wide range of texts can be effective clues for judging causality.

We extend Hashimoto et al.'s method by introducing MCNNs to effectively use a *wider* range of background knowledge expressed in a *wider* range of web texts. By "background knowledge," we refer to any type of information that is useful to generally recognize event causalities. Our assumption is that a wide range of dependency paths between nouns can be used as background knowledge, not just the short binary patterns that Hashimoto et al. proposed to exploit. In this work, we did not start from pre-specified patterns, but designed our method so that it can automatically learn/identify a wide range of paths as background knowledge in extra texts, such as why-QA system's answers. Our method is given those paths even if they are long and complex. We also even extend the notion of dependency paths to *inter-sentential* ones so that they can capture the inter-sentential relations between two nouns that appear in consecutive sentences.

In addition to the above why-QA system's answers, we tried the following two types of texts as sources of background knowledge:

A. A wider range of short binary patterns than those used in Hashimoto et al. (2014). Contrary to their work, we did not pose any semantic restrictions on the patterns.

B. One or two (consecutive) sentences that include such clue terms for causality as "reason" and "because" and the two nouns in a target event causality, like "Recently powerful *typhoons* have been reported. One *reason* is *global warming*." Note that we do not use any sophisticated mechanism such as a why-QA system to retrieve these sentences. We just retrieve all the texts including the clue terms and the nouns.

Although our target language is Japanese, we believe that our method is extendable to other languages without much cost. Note that we use English examples for the sake of readability throughout this paper.

2 Related work

For event causality recognition, researchers have exploited various clues, including discourse connectives and word sharing between cause and effect (Torisawa 2006; Abe, Inui, and Matsumoto 2008; Riaz and Girju 2010). Do, Chan, and Roth (2011) proposed cause-effect association (CEA) statistics. Radinsky, Davidovich, and Markovitch (2012) used existing ontologies, such as YAGO (Suchanek, Kasneci, and Weikum 2007). Hashimoto et al. (2012) introduced a new semantic orientation of predicates, and Hashimoto et al. (2014) exploited a handcrafted set of short binary patterns as background knowledge, as mentioned in the Introduction.

Following the seminal work of Collobert et al. (2011), convolutional neural networks (CNNs) have been applied

to such NLP tasks as document classification (Kalchbrenner, Grefenstette, and Blunsom 2014; Kim 2014; Johnson and Zhang 2015), paraphrase (Yin and Schütze 2015), and relation extraction/classification (dos Santos, Xiang, and Zhou 2015; Nguyen and Grishman 2015). Multi-column CNNs (MCNNs) were first proposed by Ciresan, Meier, and Schmidhuber (2012) for image classification. In NLP, Dong et al. (2015) used MCNNs to capture the multiple aspects of candidate answers for question-answering. Zeng et al. (2015) proposed an analogue of MCNNs called a piecewise max-pooling network for relation extraction. Our MCNN architecture was inspired by Siamese architecture (Chopra, Hadsell, and LeCun 2005), which we extend to a multi-column network and replace its similarity measure with a softmax function at its top. A similar MCNN architecture was successfully applied to zero-anaphora resolution (Iida et al. 2016).

Zeng et al. (2015) also tried to exploit external knowledge in the convolutional neural network framework. They obtained labeled training data by applying distant supervision (Mintz et al. 2009) to external knowledge, while our framework does not generate new labeled training samples. Since our additional texts are quite noisy, generating labeled samples from them would be difficult. Here we just attach those additional texts to existing labeled samples and use MCNN to identify useful background knowledge from such noisy web texts. Oh et al. (2017) proposed the most similar framework to ours; they used a variant of a multi-column convolutional neural network for why-QA and gave additional texts to columns as background knowledge.

Note that, in the previous attempts to apply MCNNs to NLP, columns were used to deal with several distinct word embeddings or distinct text fragments taken from the input texts. Our contribution is a novel way to use such MCNNs to deal with extra texts that work as a source of background knowledge for the target task.

3 Proposed method

3.1 Task definition and primary input

The primary input of our method is such event causality candidates as "smoke cigarettes" $\rightarrow$ "die of lung cancer," and our task is to judge whether they express a proper event causality. We regard causality candidate $A \rightarrow B$ proper iff "if A happens, the probability of B increases." In the candidates, the cause phrase ("smoke cigarettes") and the effect phrase ("die of lung cancer") consist of a predicate with argument position X (template, hereafter) like "smoke X" and a noun like "cigarettes" that fills X. The predicate of the cause phrase must also syntactically depend on the effect phrase, possibly through such connectives as "and" or "since" in the original sentence. The format of the event causality candidates is the same as Hashimoto et al. (2014). We chose this compact format because it contains the essence of event causalities and is easy to use in applications (e.g., future event/scenario prediction (Radinsky, Davidovich, and Markovitch 2012; Hashimoto et al. 2014)).

3.2 Method overview

In our work, the event causality candidates are fed to one of the columns in our MCNNs, as shown in Fig. 1. More precisely, the word vectors of the cause and effect parts are given to the column. Text fragments surrounding the causality candidate in its original sentence are given as contexts to the other four columns. Following the SVM-based method in Hashimoto et al. (2014), we use the following text fragments as context: (a) the text fragments between the noun and the predicate of a cause phrase, (b) the fragments between the noun and the predicate of an effect phrase, (c) the fragments between the cause noun and the effect predicate, and (d) all the words after the effect predicate. Each of the four fragments is given to a distinct column as the sequence of the word vectors. We use a total of five columns to treat the causality candidates and their contexts. In addition to the above inputs, we use extra texts as source of background knowledge as described in the next section.

3.3 Sources of background knowledge

We use three types of additional texts retrieved from our 4-billion-page web archive as background knowledge along with an event causality candidate and its contexts. In the following, we explain all of the types of text and how to feed them to our MCNNs.

Short binary patterns As a source of background knowledge for event causality recognition, Hashimoto et al. (2014) used a set of 395,578 carefully selected binary patterns, such as "A causes B" and "A prevents B," which are somehow related to event causalities.[1] The patterns here are dependency paths that connect two nouns, which are replaced with variables A and B.

We also use such binary patterns as background knowledge. Hashimoto et al. (2014) selected their patterns from a set of all the patterns in which variables A and B are filled with ten or more distinct noun pairs in 600 million web pages. This condition filters out long and uncommon binary patterns and only relatively short patterns remain. Here, we do not conduct such selection; instead we use all the patterns that survived the above ten distinct noun-pair filters. Given two nouns, the number of retrieved patterns varies from one to several hundred. Then we heuristically select a maximum of 15 binary patterns that are most frequently observed with the noun pair, concatenate them with a delimiter, "|" (e.g., "A causes B | A prevents B | ..."), and give the resulting word sequence to the sixth column in our MCNNs. The number (15) of patterns given to MCNNs was determined by preliminary experiments on our development set, in which we tried various numbers up to 250 without observing any significant changes in performance. Since a larger number extends the training time, we chose 15.

[1] Hashimoto et al. (2014) prepared the binary patterns in the following two ways: (1) by manually selecting a small number of seed patterns and expanding them by a pattern entailment database constructed by machine learning (Kloetzer et al. 2013; 2015) and (2) by adding additional predicate arguments to the unary patterns in an existing semantic predicate lexicon (Hashimoto et al. 2012; Sano et al. 2014).

Answers from a why-QA system As another source of background knowledge, we use the answers of a why-QA system, which is the implementation of Oh et al. (2013). The system retrieves passages, all of which are *seven* consecutive sentences, from four billion web pages using the terms in a given why-question and ranks the retrieved passages using a supervised ranker (SVM). The system respectively achieved accuracies of 59.3%, 83.3%, and 90.3% for the Top 1, 3, and 5 answers on our test data.[2]

In our event causality recognition, we automatically generate a question from the effect part of a given event causality candidate by basically attaching "why" ("naze" in Japanese) to its effect part. Then we retrieve the top 200 answers ranked by the why-QA system and keep those answers that contain the two nouns in the target causality candidate.

Since helpful background knowledge is often expressed as relations between the two nouns of the causality candidate, we extracted the patterns that are likely to represent such relations from the answers as follows. If the two nouns in a causality candidate appear in a single sentence, we identify the dependency paths from the individual nouns to the root of the sentence and combine them, preserving their word order and retaining the clue terms if they exist, regardless whether they are in the dependency paths. For instance, for the question, "Why (do people) die of *lung cancer*?",[3] one of the answers included the sentence, "Many people cannot stop smoking cigarettes, and, as a result, they suffer from lung cancer." From this sentence, the pattern "cannot stop A and *result* suffer from B" was extracted, where A and B are the variables for the nouns in the cause and effect parts and "result" is a clue term. If two nouns appear in consecutive sentences, we create an artificial dependency link from the root of the first sentence to that of the second sentence and extract the patterns assuming the two sentences are just one. For example, if the answer includes such sentences as "*Cigarettes* are harmful. They cause *lung cancer*, for instance," the resulting pattern is "A is harmful *causes* B" (See Fig. 2 for a Japanese example). Just like the short binary patterns, we concatenated the 15 patterns, which were extracted from the *most highly ranked answers*, with delimiters, and gave them to the seventh column in our MCNNs.

Note that the patterns extracted in this way are quite long and can give useful information that cannot be covered by short binary patterns.

One or two consecutive sentences with clue terms As a third type of knowledge source, we used sentences retrieved from four billion web pages by searching for two nouns in an event causality candidate and such clue terms as "because."[4] More precisely, we retrieved one or two consecutive sentences in which the nouns and one of the clue terms appeared. Then the patterns were extracted from the sentences by the same method as the one used for the answers from our why-QA system. We estimated the frequencies of each pattern in all the patterns extracted for our event causality training data and selected the 15 most frequent patterns for an event causality candidate. If the frequency is identical, shorter patterns are preferred. The selected patterns are concatenated with a delimiter and fed to the eighth column in our MCNNs.

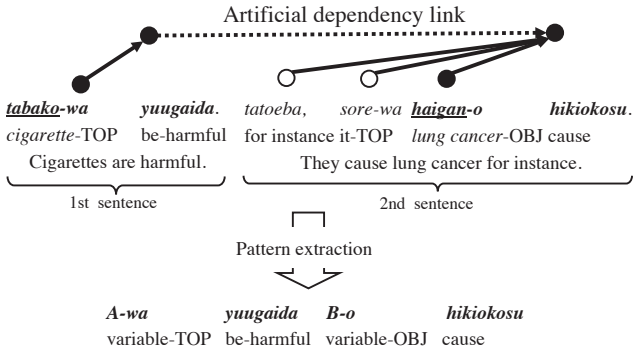

Figure 2: Example of pattern extraction from two consecutive sentences

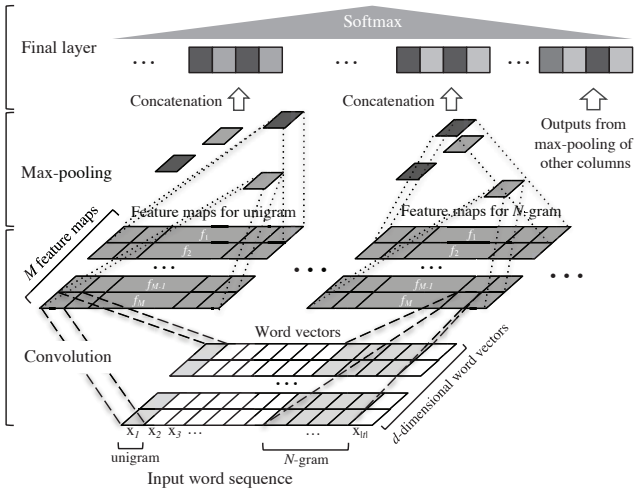

Figure 3: Column of our MCNNs

3.4 MCNN architecture

Our MCNNs consist of eight columns, one of which is shown in Fig. 3. We represent each word in text fragment t by d-dimensional embedding vector $\mathbf{x}_i$ and t by matrix $\mathbf{T} = [\mathbf{x}_1, \ldots, \mathbf{x}_{|t|}]$.[5] $\mathbf{T}$ is then wired to a set of M feature maps where each feature map is a vector. Each element O in the feature map (i.e., a neuron) is computed by a filter denoted by f_j ($1 \leq j \leq M$) from N-gram word sequences in t for some fixed integer N, as $O = \mathrm{ReLU}(\mathbf{W}_{f_j} \bullet \mathbf{x}_{i:i+N-1} + b_{f_j})$,

[2]The system was trained using 65,738 question-answer pairs for 6,109 questions. The evaluation was done against our test data that consisted of 3,438 question-answer pairs for 400 questions. Three human annotators annotated our data, and the final decision was made by a majority vote. The Fleiss' kappa was substantial (κ = 0.776).

[3]In Japanese, the subjects (e.g., people) are commonly omitted.

[4]We used 65 clue terms that were prepared manually.

[5]We used zero padding for dealing with variable-length text fragments (Kim 2014).

Data	Examples	True causalities (%)
Training	112,098	9,650 (8.6)
Dev	23,602	3,759 (15.9)
Test	23,650	3,647 (15.4)

Table 2: Statistics of our datasets

where $\bullet$ denotes element-wise multiplication followed by the summation of the resulting elements (i.e., Frobenious inner product) and $\mathrm{ReLU}(x) = \max(0, x)$. In other words, we construct a feature map by convolving a text fragment with a filter, which is parameterized by weight $\mathbf{W}_{f_j} \in \mathbb{R}^{d \times N}$ and bias $b_{f_j} \in \mathbb{R}$. Note that there can be several sets of feature maps where each set covers N-grams for different N. Since we build our MCNNs upon Siamese architecture (Chopra, Hadsell, and LeCun 2005), all the columns share the same $\mathbf{W}_{f_j}$ and b_{f_j} for the same N.

As a whole, these feature maps are called a *convolution layer*. The next layer is called a *pooling layer*. In this layer, we use max-pooling, which simply selects the maximum value among the elements in the same feature map (Collobert et al. 2011). Our expectation is that the maximum value indicates the existence of a strong clue (i.e., an N-gram) for our final judgments. The selected maximum values from all the M feature maps are concatenated, and the resulting M-dimensional vector is given to our final layer. Note that since each feature map has different weights and biases, max-pooling might select different N-gram sequences. This means we can choose multiple N-gram sequences that work as strong clues.

The final layer has vectors coming from multiple feature maps in multiple columns. They are again concatenated and constitute a high-dimensional feature vector. The final layer applies a softmax function to produce the class probabilities of the causality labels: *true* and *false*. We use a mini-batch stochastic gradient descent (SGD) with the Adadelta update rule (Zeiler 2012). We randomly initialize filter weights $\mathbf{W}_{f_j}$ from a uniform distribution in the range of $[-0.01, 0.01]$ and set the remaining parameters to zero.

4 Experiments

4.1 Settings

Hashimoto et al. (2014) extracted 2,451,254 event causality candidates from 600 million web pages. We used samples from them as our datasets. Three human annotators (not the authors) annotated the data, according to the following definition of event causality: $A \rightarrow B$ is a proper causality iff (a) "if A happens, the probability of B increases," and (b) the causality is self-contained (i.e., comprehensible without contextual information). They judged whether phrase pairs constitute a causality without their contexts. The final decision was made by a majority vote, and Fleiss' kappa showed substantial agreement ($\kappa = 0.67$). Table 2 shows the statistics of the training, development, and test data. The development and test data were randomly sampled from all the extracted candidates, but not the training set. There were no duplicate causality candidates (i.e., phrase pairs) among the three datasets.

We implemented our MCNNs using Theano (Bastien et al. 2012). We pre-trained 300-dimensional word embedding vectors using the skip-gram model (Mikolov et al. 2013) on the set of sentences (2.4M sentences, 0.6M words) from which our causality candidates were extracted. We set the skip distance to 5 and the number of negative samples to 10. We treated words that did not occur in the embedding vocabulary as unknown words. If their frequencies were less than five, we mapped all of them to a single random vector. On the other hand, if their frequencies were equal to or greater than five, we assigned each of them a distinct random vector. We also regarded the variables in the patterns as unknown words and gave them random vectors.[6] We updated all word embedding vectors during training.

To avoid overfitting, we applied early-stopping and dropout (Hinton et al. 2012). Following Graves (2013), we split the development data into two smaller sets (roughly 50%/50%) for early-stopping and selecting hyper-parameters. In all the experiments, we applied a dropout rate of 0.5 to the final layer and used an SGD with mini-batches of 100 and a learning rate decay of 0.95. We ran five epochs through all of the training data, where each epoch consisted of many mini-batch updates.

We examined the hyper-parameter settings on our development data as follows. We tried 3, 4, and 5 combinations of various N-grams where $N \in \{2, \ldots, 6\}$. One setting, for example, was (2,3,4)×200, which is interpreted as a combination of 2-, 3-, and 4-grams with 200 filters each. We restricted the N-gram combinations to consecutive numbers (e.g., the (2,3,4) combination but not (2,4,6)). The number of filters was set to 50, 100, or 200. The total possible number of hyper-parameter settings was 18; we tried all of them.

Following Hashimoto et al. (2014), we used the average precision (**AP**) as our evaluation metric. We chose the top two hyper-parameter settings by average precision in the development set, trained five models for each setting using a different random seed, and applied model averaging over 2×5 models to produce the final prediction. This strategy not only consistently improved performance but also yielded more stable results (Bengio 2012).

Table 3 presents the best hyper-parameter setting and its average precision results of our development data. **Base** is our MCNNs that use only the cause/effect phrases of a causality candidate and the contexts in the original sentence without additional background knowledge. The following are the acronyms of our knowledge sources: **BP** = short binary patterns, **WH** = why-QA system's answers, and **CL** = sentences with clue terms.

4.2 Results

Table 4 shows the experimental results on the test data, including those for our proposed methods and the other methods for comparison.

Hashimoto14 denotes Hashimoto et al. (2014)'s SVM classifier that integrates various features, such as carefully

[6] All of the random vectors were sampled from a uniform distribution in the range of $[-0.25, 0.25]$.

Method	AP
Base (2,3,4)×200	46.85
Base+BP (2,3,4,5)×100	51.29
Base+WH (3,4,5)×200	50.20
Base+CL (3,4,5)×200	49.78
Base+BP+WH (2,3,4,5,6)×200	52.23
Base+BP+CL (4,5,6)×200	52.12
Base+WH+CL (3,4,5)×200	50.65
Base+BP+WH+CL (3,4,5)×100	**52.88**

Table 3: Best hyper-parameter setting for each source combination on our development data

Method	AP
CNN-SENT	43.76
Hashimoto14	47.32
Hashimoto14+BP	47.39
Hashimoto14+WH	43.41
Hashimoto14+CL	40.11
Hashimoto14+BP+WH	47.52
Hashimoto14+BP+CL	45.96
Hashimoto14+WH+CL	41.75
Hashimoto14+BP+WH+CL	45.81
MCNN-based methods	
Base	49.34
Base+BP	54.32
Base+WH	52.03
Base+CL	52.27
Base+BP+WH	54.85
Base+BP+CL	54.36
Base+WH+CL	53.08
Base+BP+WH+CL	**55.13**

Table 4: Test data results

selected binary patterns, contexts, and association measures. We used all of their features and fine-tuned their SVM classifier on the full development data.[7] We also conducted experiments integrating our new knowledge sources to their framework to see how beneficial they are for the SVM. For instance, Hashimoto14+WH is a setting when we give the patterns from the why-QA system's answers to the SVM as binary features, like feature encoding for the short binary patterns in their method. Note that in these methods, we used all of the extracted patterns (not just the 15 patterns as in our method) for all the knowledge sources, since they used all of the patterns that they could find from web pages.

CNN-SENT denotes a CNN that contains a single column. This resembles a single-column version of our MCNNs. The difference is that it scans the original sentence, including our causality candidate. We chose its optimal hyper-parameters and performed model averaging using our described strategy.

As seen in Table 4, our proposed methods achieved significantly better average precision than the other methods. The

[7]We tried two types of kernels, including linear and polynomial (degree = 2, 3), and varied C in $\{0.0001, 0.001, \ldots, 100, 1000\}$.

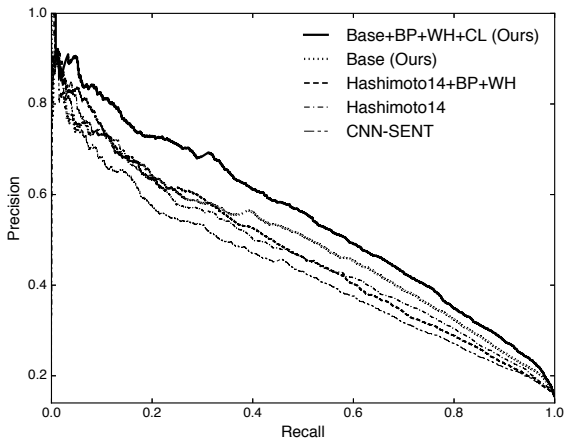

Figure 4: Precision-recall curves of our proposed methods and other methods

Method	AP
Base+BP+WH+CL	55.13
Base+BP+WH\BP+CL\BP	**55.42**

Table 5: Test data results when removing BP's patterns from WH and CL

best average precision of the proposed methods (55.13%) was achieved when we used all of the types of background knowledge (i.e., Base+BP+WH+CL), which was 7.6% higher than the best of Hashimoto et al.'s methods (47.52%). Note that we obtained 5.6% improvement by extending single CNNs to multi-column CNNs (CNN-SENT vs. Base). Integrating background knowledge further gave 5.8% improvement (Base vs. Base+BP+WH+CL). Fig. 4 shows their precision-recall curves and that our proposed methods achieved better precision than the other methods at most recall levels. These results suggest that our MCNN architecture is effective for this task.

4.3 Discussion

We further validated the contributions of the WH and CL patterns. Note that our pattern extraction method for WH and CL might produce some short binary patterns identical to those of BP, but these short patterns might not be selected as inputs to the BP column due to the limitation on the number of selected patterns (i.e., 15). We removed from WH and CL any short binary patterns that should have appeared in the input of BP's column as if there were no limitation and evaluated the resulting method. Surprisingly, removing BP's patterns from WH and CL further yielded 0.29% improvement, as shown in Table 5. This confirms the effectiveness of the complex and relatively long patterns from WH and CL.

In our proposed methods, all the knowledge sources improved the performances over Base. (See Base, Base+BP, Base+WH, and Base+CL.) This indicates that our MCNN architecture effectively exploited each type of knowledge source. The combination of different knowledge sources

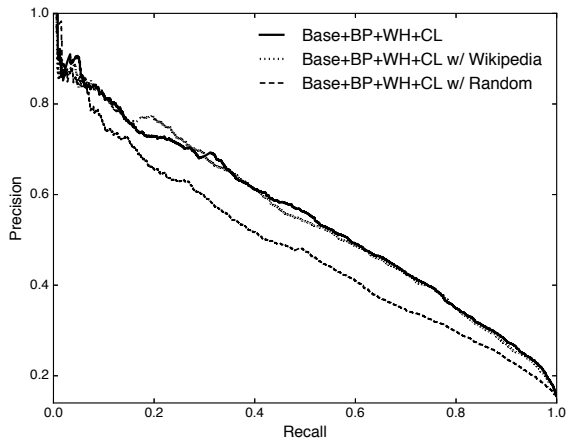

Figure 5: Precision-recall curves of our proposed methods using different word embeddings

Method	AP	
	15 ptns.	**All ptns.**
Base+BP+WH+CL (Ours)	**55.13**	-
Hashimoto14+BP+WH	47.88	47.52
Hashimoto14+BP+WH+CL	46.07	45.81

Table 6: Test data results of our method and Hashimoto et al.'s method when using identical patterns

produced better results in all cases.

The best performance (Hashimoto14+BP+WH) of Hashimoto et al.'s SVM-based method was much lower than that of our MCNN-based methods. Also, when adding our knowledge sources to Hashimoto et al.'s original method, the performance significantly dropped in most cases (e.g., Hashimoto14+WH). Such poor performance is probably due to the feature encoding scheme in their method. Their binary features for expressing patterns cannot encode explicitly semantic similarity or synonymy between distinct patterns (e.g., "A causes B" and "A is a cause of B") because the features cannot indicate that two patterns share common/synonymous words. Then the SVMs cannot capture generalization over semantically similar patterns. Contrarily, we believe that word embeddings in patterns and the (N-gram based) convolution/pooling operation on them in our method are likely to capture such similarity.

Another difference between Hashimoto et al.'s method and ours is the number of patterns (i.e., a maximum of 15 in our method and no such limitations in theirs). This may have caused overfitting in their method. We evaluated their method by posing the same limitation (Table 6). Even though the average precision was slightly improved in some cases, the best performance was only improved by 0.36% (Hashimoto14+BP+WH), suggesting that overfitting actually occurred but its effect was limited.

Finally, we examined the effects of word embedding vectors. First, we pre-trained another set of word embeddings using Wikipedia articles.[8] Second, we initialized all of the word embeddings using random initialization. Wikipedia's word embeddings gave average precision at 54.84%, while the random initialization's ones yielded much worse average precision at 48.48%. Fig. 5 compares their precision-recall curves with word embeddings trained from Hashimoto et al. (2014)'s 2.4M sentences. These results indicate that pre-trained word embeddings are one important component to the success of our proposed method. When using word embeddings trained from a more general-domain corpus like Wikipedia, our MCNN architecture still maintained relatively high average precision.

5 Conclusion

We presented a method for recognizing such event causalities as "smoke cigarettes" $\rightarrow$ "die of lung cancer." Our method exploits background knowledge extracted from noisy texts (e.g., why-QA system's answers and texts retrieved by a simple keyword search). We empirically showed that the combination of MCNNs and such background knowledge can significantly improve performance over the previous state-of-the-art method. In future work, we plan to apply our proposed method to event causality hypothesis generation (Hashimoto et al. 2015).

6 Acknowledgments

We thank Haruo Kinoshita for developing the why-QA system's API, Ryu Iida for helpful discussion, and anonymous reviewers for their insightful comments.

References

Abe, S.; Inui, K.; and Matsumoto, Y. 2008. Two-phased event relation acquisition: Coupling the relation-oriented and argument-oriented approaches. In *Proceedings of COLING*, 1–8.

Bastien, F.; Lamblin, P.; Pascanu, R.; Bergstra, J.; Goodfellow, I. J.; Bergeron, A.; Bouchard, N.; and Bengio, Y. 2012. Theano: new features and speed improvements. In *Proceedings of NIPS Workshop on Deep Learning and Unsupervised Feature Learning*.

Bengio, Y. 2012. Practical recommendations for gradient-based training of deep architectures. *Neural Networks: Tricks of the Trade* 7700:437–478.

Berant, J.; Srikumar, V.; Chen, P.-C.; Vander Linden, A.; Harding, B.; Huang, B.; Clark, P.; and Manning, C. D. 2014. Modeling biological processes for reading comprehension. In *Proceedings of EMNLP*, 1499–1510.

Chopra, S.; Hadsell, R.; and LeCun, Y. 2005. Learning a similarity metric discriminatively, with application to face verification. In *Proceedings of CVPR*, 539–546.

[8] Wikipedia articles contain 35M sentences taken from https://archive.org/details/jawiki-20150118. Base+BP+WH+CL's datasets contain 176,851 unique words. 72.61% of them can be found in the vocabulary extracted from Wikipedia articles, while 75.40% can be found from Hashimoto et al. (2014)'s 2.4M sentences.

Ciresan, D. C.; Meier, U.; and Schmidhuber, J. 2012. Multi-column deep neural networks for image classification. In *Proceedings of CVPR*, 3642–3649.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537.

Do, Q. X.; Chan, Y. S.; and Roth, D. 2011. Minimally supervised event causality identification. In *Proceedings of EMNLP*, 294–303.

Dong, L.; Wei, F.; Zhou, M.; and Xu, K. 2015. Question answering over freebase with multi-column convolutional neural networks. In *Proceedings of ACL*, 260–269.

dos Santos, C. N.; Xiang, B.; and Zhou, B. 2015. Classifying relations by ranking with convolutional neural networks. In *Proceedings of ACL*, 626–634.

Graves, A. 2013. Generating sequences with recurrent neural networks. *CoRR* abs/1308.0850.

Hashimoto, C.; Torisawa, K.; Saeger, S. D.; Oh, J.-H.; and Kazama, J. 2012. Excitatory or inhibitory: A new semantic orientation extracts contradiction and causality from the web. In *Proceedings of EMNLP-CoNLL*, 619–630.

Hashimoto, C.; Torisawa, K.; Kloetzer, J.; Sano, M.; Varga, I.; Oh, J.-H.; and Kidawara, Y. 2014. Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and assiation features. In *Proceedings of ACL*, 987–997.

Hashimoto, C.; Torisawa, K.; Kloetzer, J.; and Oh, J.-H. 2015. Generating event causality hypotheses through semantic relations. In *Proceedings of AAAI*.

Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *CoRR* abs/1207.0580.

Iida, R.; Torisawa, K.; Oh, J.-H.; Kruengkrai, C.; and Kloetzer, J. 2016. Intra-sentential subject zero anaphora resolution using multi-column convolutional neural network. In *Proceedings of EMNLP*, 1244–1254.

Johnson, R., and Zhang, T. 2015. Effective use of word order for text categorization with convolutional neural networks. In *Proceedings of NAACL-HLT*, 103–112.

Kalchbrenner, N.; Grefenstette, E.; and Blunsom, P. 2014. A convolutional neural network for modelling sentences. In *Proceedings of ACL*, 655–665.

Kim, Y. 2014. Convolutional neural networks for sentence classification. In *Proceedings of EMNLP*, 1746–1751.

Kloetzer, J.; Saeger, S. D.; Torisawa, K.; Hashimoto, C.; Oh, J.-H.; and Ohtake, K. 2013. Two-stage method for large-scale acquisition of contradiction pattern pairs using entailment. In *Proceedings of EMNLP*, 693–703.

Kloetzer, J.; Torisawa, K.; Hashimoto, C.; and Oh, J.-H. 2015. Large-scale acquisition of entailment pattern pairs by exploiting transitivity. In *Proceedings of EMNLP*, 1649–1655.

Lecun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. In *Proceedings of the IEEE*, 2278–2324.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S.; and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of NIPS*, 3111–3119.

Mintz, M.; Bills, S.; Snow, R.; and Jurafsky, D. 2009. Distant supervision for relation extraction without labeled data. In *Proceedings of ACL*.

Nguyen, T. H., and Grishman, R. 2015. Relation extraction: Perspective from convolutional neural networksperspective from convolutional neural networks. In *Proceedings of the NAACL Workshop on Vector Space Modeling for NLP*.

Oh, J.-H.; Torisawa, K.; Hashimoto, C.; Sano, M.; Saeger, S. D.; and Ohtake, K. 2013. Why-question answering using intra- and inter-sentential causal relations. In *Proceedings of ACL*, 1733–1743.

Oh, J.-H.; Torisawa, K.; Kruengkrai, C.; Iida, R.; and Kloetzer, J. 2017. Multi-column convolutional neural networks with causality-attention for why-question answering. In *Proceedings of WSDM*.

Radinsky, K.; Davidovich, S.; and Markovitch, S. 2012. Learning causality for news events prediction. In *Proceedings of WWW*, 909–918.

Riaz, M., and Girju, R. 2010. Another look at causality: Discovering scenario-specific contingency relationships with no supervision. In *Proceedings of ICSC*, 361–368.

Richardson, M.; Burges, C. J. C.; and Renshaw, E. 2013. Mctest: A challenge dataset for the open-domain machine comprehension of text. In *Proceedings of EMNLP*, 193–203.

Sano, M.; Torisawa, K.; Kloetzer, J.; Hashimoto, C.; Varga, I.; and Oh, J.-H. 2014. Million-scale derivation of semantic relations from a manually constructed predicate taxonomy. In *Proceedings of COLING*, 1423–1434.

Scaria, A. T.; Berant, J.; Wang, M.; Clark, P.; Lewis, J.; Harding, B.; and Manning, C. D. 2013. Learning biological processes with global constraints. In *Proceedings of EMNLP*, 1710–1720.

Suchanek, F. M.; Kasneci, G.; and Weikum, G. 2007. Yago: A core of semantic knowledge. In *Proceedings of WWW*, 697–706.

Torisawa, K. 2006. Acquiring inference rules with temporal constraints by using japanese coordinated sentences and noun-verb co-occurances. In *Proceedings of HLT-NAACL*, 57–64.

Yin, W., and Schütze, H. 2015. Convolutional neural network for paraphrase identification. In *Proceedings of NAACL-HLT*, 901–911.

Zeiler, M. D. 2012. ADADELTA: an adaptive learning rate method. *CoRR* abs/1212.5701.

Zeng, D.; Liu, K.; Chen, Y.; and Zhao, J. 2015. Distant supervision for relation extraction via piecewise convolutional neural networks. In *Proceedings of EMNLP*, 1753–1762.