Expert Systems With Applications 115 (2019) 512–523

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Knowledge-oriented convolutional neural network for causal relation extraction from natural language texts

Pengfei Li*, Kezhi Mao

School of Electrical and Electronic Engineering, Nanyang Technological University, 50 Nanyang Avenue, 639798, Singapore

ARTICLE INFO

Article history:
Received 26 April 2018
Revised 2 July 2018
Accepted 7 August 2018
Available online 8 August 2018

Keywords:
Natural language processing
Convolutional neural network
Relation extraction
Causal relationship
Lexical knowledge base

ABSTRACT

Causal relation extraction is a challenging yet very important task for Natural Language Processing (NLP). There are many existing approaches developed to tackle this task, either rule-based (non-statistical) or machine-learning-based (statistical) method. For rule-based method, extensive manual work is required to construct handcrafted patterns, however, the precision and recall are low due to the complexity of causal relation expressions in natural language. For machine-learning-based method, current approaches either rely on sophisticated feature engineering which is error-prone, or rely on large amount of labeled data which is impractical for causal relation extraction problem. To address the above issues, we propose a Knowledge-oriented Convolutional Neural Network (K-CNN) for causal relation extraction in this paper. K-CNN consists of a knowledge-oriented channel that incorporates human prior knowledge to capture the linguistic clues of causal relationship, and a data-oriented channel that learns other important features of causal relation from the data. The convolutional filters in knowledge-oriented channel are automatically generated from lexical knowledge bases such as WordNet and FrameNet. We propose filter selection and clustering techniques to reduce dimensionality and improve the performance of K-CNN. Furthermore, additional semantic features that are useful for identifying causal relations are created. Three datasets have been used to evaluate the ability of K-CNN to effectively extract causal relation from texts, and the model outperforms current state-of-art models for relation extraction.

1. Introduction

Relation extraction is a subfield of Information Extraction (IE) that automatically extracts semantic relations between entities, such as component-whole, product-producer, employer-employee, cause-effect, message-topic. In particular, the cause-effect relation plays an important part in human cognition due to its significant impact on reasoning and decision making (Goldvarg & Johnson-Laird, 2001; Peña, Sossa, & Gutiérrez, 2008). Causal relation extraction focuses on automatic detection of cause-effect relationship between entities in text. For example, from the sentence “the 2004 Indian Ocean earthquake triggered a series of devastating tsunamis”, we can infer causal relation between the cause “earthquake” and the effect “tsunamis”. Similarly, in “global warming resulted in higher sea levels”, the causal relation between the cause “global warming” and the effect “higher sea levels” can also be extracted. Effective extraction of causal relation from natural language texts becomes increasingly important for applications such as information retrieval (Jensen, Saric, & Bork, 2006; Khoo, Kornfilt, Oddy, & Myaeng, 1998), question answering (Girju, 2003; Higashinaka & Isozaki, 2008), event reasoning and predictions (Ackerman, 2012; Hashimoto et al., 2014; Mele & Sorgente, 2013; Radinsky, Davidovich, & Markovitch, 2012). By building causal networks (e.g. causal chain / causal map), we can induce previously unknown knowledge and apply it in various domains such as biology (Cheong & Shu, 2012), biomedicine (Kim, Ohta, & Tsujii, 2008), finance (Lee & Lee, 2012; Wang, Zhe, Kang, Wang, & Chen, 2008) and environmental science (Araúz & Faber, 2012).

Numerous efforts have been dedicated to extract causal relation from texts. The methods developed in literature fall into two categories: rule-based (non-statistical) method or machine-learning-based (statistical) method. For rule-based method, extensive manual work is required to construct handcrafted linguistic patterns including lexico-syntactic patterns and semantic patterns to infer causal relation from text based on pattern matching (Chan & Lam, 2005; Girju, 2003; Inui, Inui, & Matsumoto, 2005; Ittoo & Bouma, 2011; Khoo et al., 1998). However, regardless of extensive human effort, we cannot exhaustively conclude all the linguistic patterns of causal relation due to the complexity of causal relation expressions in natural language, including lexical, morphological and syntactic variations. Besides, the polysemous patterns with

* Corresponding author.
E-mail addresses: pli006@e.ntu.edu.sg (P. Li), ekzmao@ntu.edu.sg (K. Mao).

https://doi.org/10.1016/j.eswa.2018.08.009

exact pattern matching yield low precision and recall due to lexical ambiguities. For machine-learning-based method, causal relations are automatically inferred from large amount of labeled data. In early years, people construct rich features (lexical, syntactic or semantic features) by sophisticated feature engineering, where the handcrafted features are carefully designed and adjusted for specific patterns and domains (Bethard & Martin, 2008; Pakray & Gelbukh, 2014; Sorgente, Vettigli, & Mele, 2013; Yang & Mao, 2014). The performance of the system strongly depends on the quality of the designed features. Particularly, many features require external NLP toolkits such as POS tagger, dependency parser, named entity recognizer and sentence parser. Many NLP toolkits are imperfect and will cause error propagation to the causal relation extraction system.

Recently, with the prevalence of deep learning (Goodfellow, Bengio, Courville, & Bengio, 2016), researchers started to construct models without complicated feature engineering and to minimize the reliance on NLP toolkits for feature acquisition. One of the most prominent deep learning model for relation extraction is Convolutional Neural Network (CNN) with pre-trained word embeddings, which achieved state-of-art performance for SemEval-2010 task 8 challenge (Nguyen & Grishman, 2015; Santos, Xiang, & Zhou, 2015; Zeng, Liu, Lai, Zhou, & Zhao, 2014). The pre-trained word embeddings encode semantic and syntactic information of words into fixed-length vectors (Mikolov, Chen, Corrado, & Dean, 2013; Turian, Ratinov, & Bengio, 2010) and CNN is able to extract meaningful n-grams from sentences (Kalchbrenner, Grefenstette, & Blunsom, 2014). Compared with rule-based methods and rich features-based methods, the CNN models with word embeddings are able to extract complex causal relations more effectively. However, these models rely on large amount of training data which should cover all the causal relation expressions in natural language. This is impractical due to the variety and ambiguity of words and sentences in natural language. With a large number of free parameters in deep learning models, the models can easily over-fit the limited and biased training data, and thus hinder the performance of deep learning models.

In this paper, we propose a novel deep learning model called Knowledge-oriented Convolutional Neural Network (K-CNN) which is specially tailored for causal relation extraction. The model combines human prior knowledge and the information learned from data in a complementary way to extract causal relation from natural language texts. The knowledge-oriented channel of K-CNN incorporates the linguistic knowledge of causal relationship from lexical knowledge bases including FrameNet and WordNet to capture the significant linguistic clues of causal relationship. In conventional CNN, the weights of convolutional filters are trained using training data via back-propagation, whereas the convolutional filters of knowledge-oriented channel are automatically generated from lexical knowledge bases to represent keywords and cue phrases of causal relations, which are called word filters in this paper. The data-oriented channel of K-CNN uses conventional CNN which gives the model enough capacity to adjust itself to learn other important features of causal relationship from training data. The weights of the word filters in knowledge-oriented channel are equal to the pre-trained word embeddings and are kept fixed, hence the number of free parameters of the model is significantly reduced. We also propose word filter selection and clustering techniques to remove non-discriminative and redundant features. Additionally, two types of semantic features including WordNet categorical features and FrameNet causal scores are created to capture useful semantic information hidden in the texts.

The main contributions of our work are summarized as follows:

- We propose an effective way of combining human prior knowledge and information from data for convolutional neural network. The convolutional filters in knowledge-oriented channel of K-CNN are automatically generated from lexical knowledge bases. Compared with conventional CNN, the proposed word filters are able to represent keywords and cue phrases of causal relationship more accurately. The specially designed convolutional operations allow the model to extract these significant clues of causality from texts and output a similarity score for each word filter, this overcomes the problem of rigid pattern matching for rule-based methods. As word filters are static, the number of free parameters in K-CNN is significantly reduced compared with conventional CNN. Hence, overfitting issue can be alleviated and training efficiency can be improved.
- The proposed word filter selection and clustering techniques for dimensionality reduction can further improve the performance of K-CNN. Firstly, the insignificant word filters are removed using hypothesis testing based on Analysis of Variance (ANOVA) F-ratio; Then the redundant features are removed by clustering of the remaining word filters based on their semantic similarities. The convolutional results for word filters within each cluster are grouped together by using either average-pooling or max-pooling, the performances of the two pooling methods are investigated.
- Two types of semantic features including WordNet categorical features and FrameNet causal scores are proposed, these features are able to discover more semantic information of causal relationship hidden in the text, which allow our model to better deal with complex and implicitly expressed causal relations.
- Comprehensive experiments are conducted on three datasets for causal relation extraction. The experiment results demonstrate that our proposed K-CNN is able to extract causal relations more effectively than conventional CNN models.

This paper is organized as follows. In Section 2, some related works using machine-learning-based method are reviewed. Our proposed Knowledge-oriented CNN (K-CNN) for causal relation extraction is presented in Section 3. Experimental results and analyses are presented in Section 4. Finally, our work as well as future work are summarized in Section 5.

2. Related work

As the performance advantage of statistical methods over non-statistical methods for causal relation extraction in the literature, we concentrate on statistical methods in this section. To be more specific, we focus on supervised systems including basic machine-learning models as well as deep neural network models. The difference of the two types of models is that sophisticated feature engineering is normally required for basic machine-learning models such as Bayes classifier, Support Vector Machine (SVM) and Decision Tree; whereas for deep neural network models, features are automatically learned from training data.

2.1. Models with sophisticated feature engineering

There are two main steps for supervised machine-learning-based approaches, including feature extraction/generation and pattern classification. Various methods were proposed for both steps in the literature.

Girju, Moldovan et al. (2002) firstly generate syntactic patterns in the form of $\langle NP1\ causal - verb\ NP2 \rangle$ by searching a collection of texts on the Internet, where $NP1$ and $NP2$ are found from lexical knowledge base WordNet. Then, the extracted patterns are validated and ranked by imposing some constraints on *NP1, NP2* and *causal-verb* through a WordNet-based course-grained process. Their work was modified in Girju (2003) where the semi-automatic pattern validation process is replaced by a supervised method using C4.5 decision tree. Even through the causal relation is restricted to noun pairs with certain causal verbs as the connector, Girju and Moldovan's work demonstrates that machine-learning framework is more effective than conventional rule-based framework. Instead of using large amount of causally-labeled corpus, which is time and effort consuming, Chang and Choi (2004) generate ternary patterns ⟨*cause NP, cue phrase, effect NP*⟩ from dependency structure of sentences. The cue phrase probability and lexical pair probability of the ternary patterns are generated from unlabeled corpus by using EM (Expectation-Maximization) procedure. These probabilities are used as features and Naive Bayes classifier is used to classify the pattern as causal or non-causal. In Blanco, Castell, and Moldovan (2008), seven types of features are constructed, including the types of relator, left and right modifiers of the relator, semantic classes of cause verb (as well as effect verb), whether the cause verb (as well as effect verb) is potentially causal, and the tense of the cause and effect verbs. With these features, Bagging with C4.5 decision trees classifier is used to make binary classification, and a satisfying classification result is achieved. However, the model considers only causal relations with a fixed pattern ⟨*VerbPhrase Relator Cause*⟩ containing only four relator types: *after, as, because* and *since*, which is too simplified for causal relationship.

With the development of benchmarked dataset for semantic analysis tasks in NLP, such as SemEval-2007 task 4 and SemEval-2010 task 8 for relation classification, the semantic relations analyses including causal relation extraction have been improving and more novel methods have been developed. The winners of the two tasks ((Girju, Beamer, Rozovskaya, Fister, & Bhat, 2010) for SemEval-2007 task 4 and (Rink & Harabagiu, 2010) for SemEval-2010 task 8) both use a combination of lexical, syntactic and semantic features extracted from various knowledge resources (e.g. WordNet) and NLP toolkits (e.g Syntactic Parser), and Support Vector Machines (SVMs) as the classifier. Although the classification results for Cause-Effect relation are satisfying, most samples in the datasets are simple causal relations which are explicitly expressed with significant linguistic clues (such as "cause", "result in", "because"), the extraction of complex and implicitly expressed causal relations is still a challenging task.

Recently, many works are proposed to handle more complex causal relations. Rink, Bejan, and Harabagiu (2010) proposed a novel graph pattern based framework. The lexical, syntactic, and semantic features of a sentence are automatically extracted and encoded into a graph representation which shows the dependency between features. The most representative graph patterns for causal relationship are discovered and the sentence is classified based on these pattern features using SVM. Sorgente et al. (2013) combined both rules and machine-learning methods for causal relation extraction. The rules are defined based on lexico-syntactic patterns and dependency structure of the sentence to extract possible cause-effect pairs from a sentence, then a Bayesian classifier with Laplace smoothing is used to discard incorrect pairs. Yang and Mao (2014) proposed a multi-level relation extraction algorithm (MLRE) to extract all potential causal relations with any verb or preposition based on the linguistic structures of dependency grammar and constituent grammar. Informative features are found based on lexical knowledge bases (WordNet, VebNet and FrameNet) and feature selection technique. Ensemble learning using restricted boosting and SVM with RBF kernel are used to make classification. Zhao, Liu, Zhao, Chen, and Nie (2016) proposed a Restricted Hidden Naive Bayes model based on causal connectives analysis.

Despite the significant improvements made by basic machine-learning models over rule-based models, there are still many problems in current systems. Firstly, sophisticated feature engineering is required as the model rely on a suitable feature set with good separation capability, which is labor and time consuming and hard to achieve. Secondly, many features are extracted based on existing NLP toolkits, such as POS tagger, dependency parser, named entity recognizer etc. Some NLP toolkits are imperfect and the errors caused by these toolkits will propagate to the causal relation extraction system. Thirdly, many systems are domain-dependent and the features need to be re-designed to adopt other domains.

2.2. Deep neural network models

The goal of constructing deep neural network models is to allow the model to learn and extract useful features automatically and more effectively than handcrafted features designed by human. In NLP, such models are primarily based on a distributed representation of words in vector space, called word embeddings, which is capable of capturing semantic and syntactic properties of words (Mikolov et al., 2013; Turian et al., 2010). Among various deep neural network models, two most widely used models for relation classification are recursive neural networks (RNNs) and convolutional neural networks (CNNs).

Socher, Huval, Manning, and Ng (2012) presented a novel method for relation classification using recursive neural network (RNN) to learn compositional vector representations for phrases and sentences through the syntactic tree path that connects two nominals, and applied these representations to classify the semantic relationship between them. Hashimoto, Miwa, Tsuruoka, and Chikayama (2013) also used an RNN for relation classification, the novelty is that their method allows the explicit weighting of important phrases for the target task. However, RNN-based methods inherently require syntactic parse trees which may introduce error propagation.

Convolutional neural network (CNN) is a special feed-forward neural network whose layers are formed by convolution operations followed by pooling operations (LeCun, Bottou, Bengio, & Haffner, 1998). Recently, CNN has demonstrated its advantage in capturing semantic and syntactic information of n-grams and inducing more abstract and discriminative representations of textual inputs (Kalchbrenner et al., 2014; Kim, 2014). Based on the idea of modeling sentences using CNN, many works have been proposed for relation extraction and achieved start-of-art performance. Zeng et al. (2014) first used position embeddings together with pre-trained word embeddings for relation classification; Nguyen and Grishman (2015) improved their model by using convolutional filters with multiple window sizes and fine-tuning the pre-trained word embeddings; Instead of using CNN followed by a softmax classifier, Santos et al. (2015) proposed a classification by ranking CNN (CR-CNN) that learns a distributed vector representation for each class by minimizing a novel pairwise ranking loss function. Lee, Dernoncourt, and Szolovits (2017) further added entity type and part of speech information into the embedding and used rule-based post-processing to correct the relations detected by CNN, their system ranked first in SemEval-2017 task 10 for relation extraction in scientific articles.

Despite the better performance of deep neural network models, the models normally require large amount of labeled data to learn, and it is impractical to construct sufficient training data that covers all causal relation expressions in natural language. With a large number of free parameters in deep learning model and insufficient or biased training data, the model can easily over-fit the training data, resulting bad generalization on test data. There are some recent works (Gómez-Adorno, Posadas-Durán, Sidorov, & Pinto, 2018; Posadas-Durán et al., 2017) that learn distributed document em-

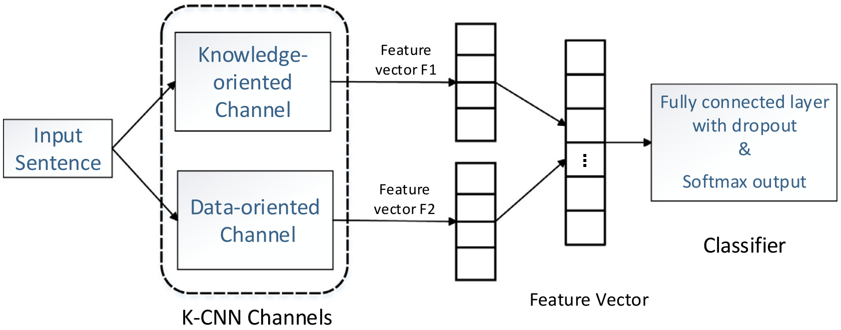

Fig. 1. Overall architecture of K-CNN.

beddings based on unsupervised context modeling, called Paragraph Vector (Le & Mikolov, 2014). The models require less training data and able to capture word semantics and n-gram information from texts as well. The resulted document embeddings have the property that the documents containing semantically similar n-grams have closer embeddings in vector space. However, the embeddings learned in this way may not be suitable to be used for causal relation extraction directly, as sentences with semantically similar n-grams may or may not involve causal relation. For example, "the *tsunami* generated by *earthquake* killed hundreds of thousands of people" involves causal relation between "tsunami" and "earthquake", whereas "the *tsunami* and *earthquake* killed hundreds of thousands of people" does not. Our work attempts to automatically construct the convolutional filters of CNN based on the linguistic knowledge of causal relation from lexical knowledge bases, which allows the model to extract significant clues of causality from natural language texts effectively and precisely, and this alleviates the overfitting issue of deep learning models.

3. Knowledge-oriented CNN (K-CNN)

In this section, our proposed Knowledge-oriented Convolutional Neural Network (K-CNN) is presented in detail. The overall architecture of our proposed K-CNN is shown in Fig. 1. The model combines human prior knowledge and information from data to extract causal relationship. It consists of two CNN channels, one is called knowledge-oriented channel, which incorporates existing linguistic knowledge from lexical knowledge bases; the other is called data-oriented channel, which learns important features from data. The two channels complement with each other and extract useful features of causal relation from different perspective.

3.1. Knowledge-oriented channel

The knowledge-oriented channel is the main part of K-CNN, which effectively extracts keywords and cue phrases of causal relationship from sentence. The convolutional filters in this channel are automatically generated based on the linguistic knowledge of causal relationship in lexical knowledge bases including WordNet[1] and FrameNet[2], we call them word filters. Compared with the convolutional filters in conventional CNN, word filters are able to represent linguistic clues of causal relationship more precisely. More importantly, the weights of word filters are embeddings of the words, which are pre-trained and directly used without any additional training. Hence, the number of free parameters of the model can be significantly reduced, alleviating the overfitting issue when the amount of training data is small. The detailed architecture of the knowledge-oriented channel of K-CNN is shown in Fig. 2.

3.1.1. Sentence representation

The input to K-CNN is a sentence marked with two target entities e_1 and e_2 for causal relation identification. It is observed that majority of keywords and cue phrases of causal relationship such as "cause", "lead to", "result in" appear in the words between e_1 and e_2, and causal relations can be identified by looking into these words. Keywords and cue phrases appear far away from two target entities may not be informative and will affect the classification result, as illustrated in the examples in Section 3.2.1. To remove noise and effectively extract these linguistic clues from the sentence, we only use words between two target entities as input to the knowledge-oriented channel. To reduce morphological variations of words and make them consistent with WordNet tokens, each word is transformed into its lower case and base form using WordNet lemmatizer. To capture the syntactic and semantic information of words, each word is represented by a vector $\mathbf{w} \in \mathbb{R}^e$ by looking up the word embedding table $\mathbf{W^{wrd}} \in \mathbb{R}^{e \times |V|}$ that is pre-trained using large corpus, where e is the dimensionality of word embedding vectors and $|V|$ is the vocabulary size. As CNN only works with fixed length inputs, the number of word tokens in a sentence is fixed at n_1 which is the maximum number of words between e_1 and e_2. The sentences with less than n_1 tokens are padded using a special padding character with zero embedding vector. Hence, the input $x_K = \left\{x_1, x_2, \ldots, x_{n_1}\right\}$ is represented as a sequence of real-value vectors $\mathbf{emb_K} = \left\{\mathbf{w_1}, \mathbf{w_2}, \ldots, \mathbf{w_{n_1}}\right\}$.

3.1.2. Automatic word filter bank generation

This process automatically generates convolutional filters of CNN for causal relation extraction without training the model using large amount of data. The filters are actually the embeddings of causation words, which are extracted from two publicly available lexical knowledge bases, WordNet and FrameNet.

WordNet is a large lexical database that groups English words into sets of synonyms (called synsets) to represent different meanings or concepts (Fellbaum, 2010). The synsets are connected in a hierarchical structure to reflect their lexical and semantic relations, and the meaning of each synset is explained by a gloss with some examples. The example below shows the WordNet elements of one of the synsets for the word "cause".

- **Synset members**: cause, do, make.
- **Gloss**: give rise to; cause to happen or occur, not always intentionally.
- **Example**: "cause a commotion"; "make a stir"; "cause an accident".
- **Direct troponym**: determine, initiate, effect, make, occasion, provoke, motivate...
- **Direct hypernym**: make, create.

[1] Available online at https://wordnet.princeton.edu.
[2] Available online at https://framenet.icsi.berkeley.edu/fndrupal.

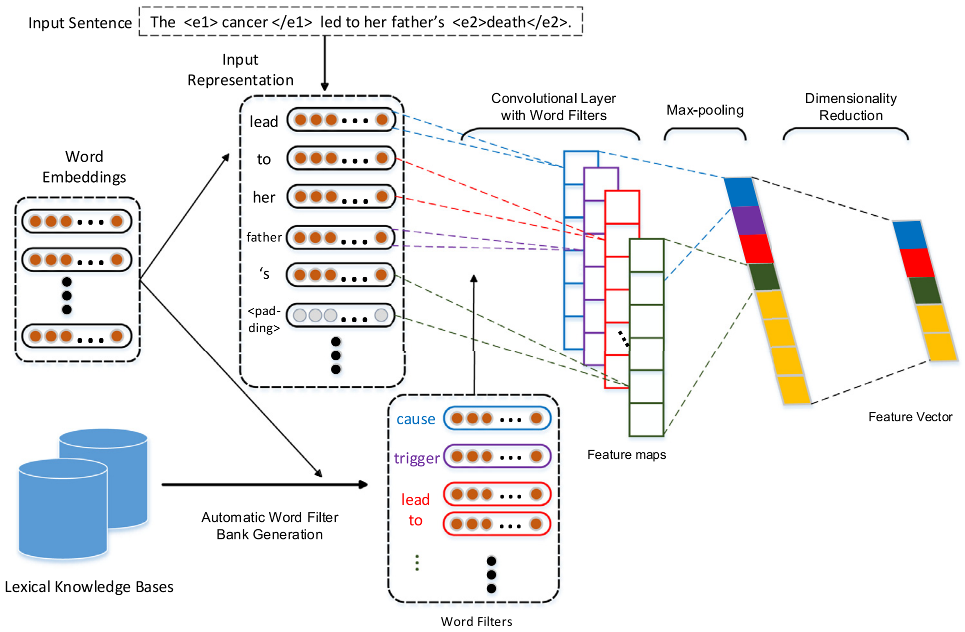

Fig. 2. Knowledge-oriented channel of K-CNN.

- **Derivationally related form**: causative, causation, cause.
- **Sentence frame**: Somebody —s something; Something —s something.

FrameNet is a lexical resource built on the frame semantic theory, and it categorizes English words and sentences into higher level semantic frames describing different concepts (Ruppenhofer, Ellsworth, Petruck, Johnson, & Scheffczyk, 2016). Each frame is a conceptual structure describing a type of event, relation or object, which is provided with conceptual definition, elements participated in the frame (frame elements), words that often appear in the frame (lexical units) and relationship with other frames. The example below shows the FrameNet elements of the “Causation” frame.

- **Definition**: A Cause causes an Effect. Alternatively, an Actor, a participant of a (implicit) Cause, may stand in for the Cause. The entity Affected by the Causation may stand in for the overall Effect situation or event.
- **Frame Elements (FEs)**:
  - Core: Actor, Affected, Cause, Effect.
  - Non-core: Circumstances, Concessive, Explanation, Frequency, Manner, Means, Place, Time.
- **FE core sets**: {Actor, Cause}, {Affected, Effect}.
- **Examples**: “He MADE me angry.”, “If such a small earthquake CAUSES problems, just imagine a big one!”, “The strange mutations of the rumor mill in the end LED to it being said that he was actually a woman.”...
- **Lexical units**: because of, cause, consequence, due to, for, force, lead (to), result (in), since...
- **Frame-frame relations**:
  - Inherits from: Eventive_affecting
  - Is inherited by: Cause_to_start
  - Perspective on: Causation_scenario
  - Is perspectivized in: Means
  - Is used by: Level_of_force_exertion

FrameNet contains over 1200 semantic frames, 13,000 lexical units and 202,000 example sentences, some of the semantic frames can be used as the evidence for causal relation (Riaz & Girju, 2014), we refer them as “causal frames”. In our work, we identify 40 causal frames from FrameNet including “Causation”, “Causation_Scenario”, “Triggering”, “Reason”, “Explaining_the_facts”, “Response” as well as 34 frames start with “Cause”. The lexical units involved in these causal frames are the significant clues and frequently appeared words that invoke causality in the sentence, hence these lexical units can be treated as keywords and cue phrases for causal relations. We further extend these lexical units using WordNet for a broader coverage of causal words and construct a bank of causal words automatically. The weights of the convolutional filters of CNN (word filters) are found based on these causal words and word embeddings. The details of the algorithm for automatically generating word filter bank are described in Algorithm 1.

The word filters constructed in this way have physical meanings that represent keywords and cue phrases of causal relationship. These word filters are constructed based on human prior knowledge of causal relationship, they are more precise than conventional convolutional filters that are learned from training data. Moreover, the weights of these filters are static values instead of free parameters in the model. Hence, the number of free parameters in the model is significantly reduced, alleviating the overfitting issue when the amount of training data is small. Eventually, around 650 uni-gram, 140 bi-gram and 10 tri-gram word filters are generated from lexical knowledge bases.

Besides WordNet and FrameNet, there are a few other popular lexical knowledge bases such as PropBank (Kingsbury & Palmer, 2003), VerbNet (Schuler, 2005) and OntoNotes (Hovy, Marcus, Palmer, Ramshaw, & Weischedel, 2006), which can also be utilized to extract the linguistic knowledge of causal relation. Both PropBank and VerbNet can be used to find the keywords of causation. However, unlike FrameNet which is semantically motivated and contains lexical units with various part of speeches, PropBank and VerbNet are verb-oriented and focus more on syntactic level. Hence, many important linguistic clues of causal relation such as “because”, “since”, “from”, “due to”, “result in” cannot be extracted, and many verbs with no causal meaning may be extracted unexpectedly, producing more noises to the causal relation extraction system. OntoNotes sense groupings can be used to extend the causal words found from FrameNet. However, word senses in

Algorithm 1 Automatic word filter bank generation.

```
Step 1: Find all the lexical units of 40 causal semantic frames from FrameNet, group them according to No. of words (max: 3).
  $lu_1 = \{c_1, c_2, \dots, c_{n_1}\}$
  $lu_2 = \{[c_{11}, c_{12}], [c_{21}, c_{22}], \dots, [c_{n_2 1}, c_{n_2 2}]\}$
  $lu_3 = \{[c_{11}, c_{12}, c_{13}], [c_{21}, c_{22}, c_{23}], \dots, [c_{n_3 1}, c_{n_3 2}, c_{n_3 3}]\}$
Step 2: Extend lexical units of $lu_1$ using WordNet.
  for word in $lu_1$ do
    for synset in WordNet synsets of word do
      if {"cause","effect","causal","causation","result", "reason","because", "responsible"} in WordNet gloss of synet then
        for lemma in WordNet lemmas of synet do
          if length of lemma == 1 then
            $lu_1 = lu_1 + lemma$
          else if length of lemma == 2 then
            $lu_2 = lu_2 + lemma$
          else if length of lemma == 3 then
            $lu_3 = lu_3 + lemma$
          end if
        end for
      end if
    end for
  end for
Step 3: Generate CNN convolutional filter weights.
  for each lexical unit $[c_1, \dots, c_k]$ in $lu_k$, $(k = 1, 2, 3)$ do
    the corresponding filter weights is:
      $\mathbf{f} = [\mathbf{f_1}, \dots, \mathbf{f_k}]^T$
    where $\mathbf{f_i} \in \mathbb{R}^e$ is the word embedding of $c_i$ found by looking up the word embedding table $\mathbf{W^{wrd}} \in \mathbb{R}^{e \times |V|}$, and $k$ is the convolutional window size.
  end for
```

OntoNotes are more coarse-grained compared with WordNet, resulting in many irrelevant words.

3.1.3. Convolution and pooling operations

To capture the significant linguistic clues of causal relationship in the sentence, the word filters generated by Algorithm 1 are convolved with the n-grams in the sentence to produce a sequence of similarity scores. Compared with rule-based pattern matching, the proposed convolutional approach is able to capture semantically similar causal words other than the words in the word filter bank.

To be more specific, the word filters $\mathbf{f} = [\mathbf{f_1}, \dots, \mathbf{f_k}]^T$ generated by Algorithm 1 are convolved with the input matrix $\mathbf{emb_K} = \{\mathbf{w_1}, \mathbf{w_2}, \dots, \mathbf{w_{n_1}}\}$, where $k \in [1, 2, 3]$ is the convolutional window size which also represents the k-grams in the sentence. We modified the conventional convolutional operation of CNN to make each word filter generates a feature map $\mathbf{m} = [m_1, m_2, \dots, m_{n_1-k+1}]$, where m_i represents the similarity between the word filter $\mathbf{f}$ and the k-gram $\mathbf{w}_{kgram} = [\mathbf{w_i}, \dots, \mathbf{w_{i+k-1}}]^T$ in the sentence. The modified convolutional operation is expressed in Eq. (1).

$$m_i = \left(\sum_{j=1}^{k} \mathbf{f}_j^T \mathbf{w}_{i+j-1} + b \right) / k \tag{1}$$

where b is a bias term. Unlike conventional CNN of applying a nonlinear function to the convolution result, we divide the convolution result by the window size k instead. By restricting $\mathbf{f}_j$ and $\mathbf{w}_{i+j-1}$ (word embeddings) as unit vectors, the resulting value of m_i becomes cosine similarity between $\mathbf{f}$ and $\mathbf{w}_{kgram}$. The mathematical proof is omitted due to limited space. The reason why we produce cosine similarity in the feature map is that we want equal scale for all convolutional window sizes, hence achieving equal importance for word filters with different lengths. Whereas conventional convolution operation will result in higher value for longer window size.

We use max-pooling to further aggregate the convolution results for each filter and extract the most significant or relevant feature from the feature map. The max-pooling operation for each feature map is expressed in Eq. (2).

$$p = max\{\mathbf{m}\} = max\{m_1, m_2, \dots, m_{n_1-k+1}\} \tag{2}$$

The rationale for taking the maximum value from feature map is that the largest cosine similarity indicates strong clue of the existence of causal keywords or cue phrases in the sentence.

3.1.4. Word filter selection and clustering

There are nearly 800 word filters generated from Algorithm 1, which is considered as high dimensionality compared with limited training data, and many features generated from these word filters may be irrelevant or redundant for causal relation classification task. To boost the performance of the model, word filter selection and clustering techniques are proposed for dimensionality reduction.

Word filter selection

We first remove non-discriminative features produced from the word filters which do not provide enough class separability. Based on the training data and their labels, we use analysis of variance (ANOVA) F-ratio to evaluate the separability of each feature, which measures the degree of the difference among class means (David, 2018). If the F-ratio is very small, then the mean of the classes are almost the same and the feature is not helpful for classification; If the F-ratio is large, then at least one class mean is different from others, which means the feature is able to provide useful information for the classifier to distinguish this class from others. The F-ratio for each feature generated after max-pooling layer is calculated by taking the ratio of mean square between classes (MSB) and mean square within classes (MSE) as shown in Eq. (3).

$$\begin{aligned} F &= \frac{MSB}{MSE} = \frac{SS_{condition}/df_n}{SS_{error}/df_d} \\ &= \frac{\left[\sum_{i=1}^{c} n_i (M_i - GM)^2\right]/(c-1)}{\left[\sum_{i=1}^{c} \sum_j (x_{ij} - M_i)^2\right]/(N-c)} \end{aligned} \tag{3}$$

where SS is sum of squares, df is degree of freedom, c is the number of classes (conditions), N is the number of all samples, n_i is the number of samples in class i, GM (grand mean) is the mean of all samples, M_i is the mean of class i, and x_{ij} is the j-th sample in class i.

After calculating the F-ratio for each feature, hypothesis testing is performed based on the F-distribution of the F-ratio with degree of freedom $c - 1$ and $N - c$. The null hypothesis H_0 assumes that all class means are equal. We take a significance level of $\alpha = 5\%$ and the critical F-ratio F_α can be found from F-distribution. If $F > F_\alpha$, H_0 can be rejected and the corresponding filter is kept because it indeed provides separability of the class means; if $F \le F_\alpha$, the corresponding filter is removed.

Clustering of word filters

After removing non-discriminative features, there are still many redundant features, which produce same or close values. This will increase the dimensionality of the model and may hurt the classification performance. These redundant features are produced by semantically similar word filters which have close word embeddings. Intuitively, we perform clustering of word filters based on their semantic similarities to solve the feature redundancy problem.

As there are only 10 tri-gram word filters generated, which is a very small number, we only cluster the uni-gram and bi-gram word filters. For bi-gram word filters, we concatenate two word embeddings to form a single vector representation. Then K-means clustering algorithm is performed to find the clusters of uni-gram word filters and bi-gram word filters separately. Based on these clusters, a further pooling operation is performed on the features after the max-pooling layer of CNN. For features $\{p_{i1}, p_{i2}, \ldots, p_{in}\}$ which correspond to the word filters within the i-th cluster, max-pooling or average-pooling is performed in the following way:

$$q_i = \begin{cases} max\{p_{i1}, p_{i2}, \ldots, p_{in}\} & \text{for max-pooling} \\ (\sum_{j=1}^{n} p_{ij})/n & \text{for average-pooling} \end{cases} \quad (4)$$

The max-pooling operation preserves the most significant linguistic clues in the sentence, whereas the average-pooling operation takes into consideration of all the word filters.

As the pooling operation aggregates the features within one cluster into a single feature, the dimensionality of the feature vector is reduced. The number of clusters of uni-gram and bi-gram word filters, h_1 and h_2, which can be set by users, determines the dimensionality of the feature vector from knowledge-oriented channel, which is equal to $h = h_1 + h_2 + 10$.

3.2. Data-oriented channel

The data-oriented channel of K-CNN uses conventional CNN to learn important features of causal relationship from the training data. Compared with knowledge-oriented channel, it captures longer dependencies in the whole sentence by using longer convolutional window sizes. By allowing the convolutional filters to adjust their weights based on the training data, we give the model enough capacity to learn such longer dependencies and important information that is neglected by the knowledge-oriented channel. Hence, data-oriented channel plays a complementary role to knowledge-oriented channel, and the combination of the two channels allows K-CNN to extract causal relation from sentence effectively.

3.2.1. Sentence representation

To preserve the information in the words besides words between e_1 and e_2, the input to the data-oriented channel contains all the words in the sentence. In this case, the maximum number of words in the sentence (n_2) can be very large compared with n_1. However, CNN is not able to capture the position information of words in the sentence, and words appear far away from e_1 and e_2 may not be informative. In the examples below, CNN is able to capture the cue phrase “result in” and correctly classify the relationship between e_1 and e_2 as causal relation in sentence 1. However, CNN may still classify the relationship between e_1 and e_2 in sentence 2 as causal by mistake once it captures the same cue phase “result in” without any position information. Hence, position information is needed for CNN to deal with the semantic analysis of long sentences.

1. *The financial ⟨e_1⟩crisis⟨/e_1⟩ **resulted in** 12% ⟨e_2⟩unemployment⟨/e_2⟩ in this country.*
2. *In spite of great ⟨e_1⟩effort⟨/e_1⟩ put by the ⟨e_2⟩government⟨/e_2⟩ in providing more employment opportunities for low incomes, the financial crisis still **resulted in** 12% unemployment in this country.*

We adopt the word position embeddings proposed by Zeng et al. (2014) to enable the CNN to keep track of how close the words are to e_1 and e_2. For each word x_i in the input sentence, its relative distances to e_1 and e_2 ranging from $1 - n_2$ and $n_2 - 1$ are calculated first. For example, in sentence 1, the word “resulted” has relative distances of 1 and -3 to e_1 and e_2 respectively. Then the two relative distances are mapped into real-value vectors $\mathbf{p_i^1}, \mathbf{p_i^2} \in \mathbb{R}^d$ by looking up the position embedding table $\mathbf{W^{psn}} \in \mathbb{R}^{d \times (2n_2 - 1)}$ which is initialized randomly, where d is a hyper-parameter indicating the dimensionality of each position embedding, and $(2n_2 - 1)$ is the total number of possible relative distances. Finally, the word embedding of x_i, $\mathbf{w_i}$ is concatenated with $\mathbf{p_i^1}$ and $\mathbf{p_i^2}$ as $\left[\mathbf{w_i}, \mathbf{p_i^1}, \mathbf{p_i^2}\right]^T$. As a result, the input $x_D = \left\{x_1, x_2, \ldots, x_{n_2}\right\}$ is represented as a sequence of real-value vectors $\mathbf{emb_D} = \left\{\left[\mathbf{w_1}, \mathbf{p_1^1}, \mathbf{p_1^2}\right]^\mathbf{T}, \left[\mathbf{w_2}, \mathbf{p_2^1}, \mathbf{p_2^2}\right]^\mathbf{T}, \ldots, \left[\mathbf{w_{n_2}}, \mathbf{p_{n_2}^1}, \mathbf{p_{n_2}^2}\right]^\mathbf{T}\right\}$.

3.2.2. Convolution and pooling operation

We use conventional CNN in data-oriented channel. The convolutional filters $\mathbf{f} = [\mathbf{f_1}, \mathbf{f_2}, \ldots, \mathbf{f_k}]^T$ are initialized randomly and trained through back-propagation, where $\mathbf{f_i} \in \mathbb{R}^{e+2d}$ and k is the convolutional window size. To capture longer dependencies in the sentence instead of causal keywords only, we use wider window size ($k = 3, 4$) compared with knowledge-oriented channel.

We simplify the representation of input sentence as $\mathbf{emb_D} = \{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_n}\}$. The convolution operation in this channel is shown in Eq. (5).

$$m_i = tanh\left(\sum_{j=1}^{k} \mathbf{f}_j^T \mathbf{x}_{i+j-1} + b\right) \quad (5)$$

where $tanh$ is hyperbolic tangent function, and b is a bias term. The resulting feature map $\mathbf{m} = \left[m_1, m_2, \ldots, m_{n-k+1}\right]$ is passed to the same max-pooling layer as in knowledge-oriented channel to extract the most significant feature. The number of filters r, which is a hyper-parameter chosen by the user, determines the output dimensionality of data-oriented channel.

3.3. Additional features for causal relation extraction

For causal relations with specific linguistic clues such as “cause”, “because”, “due to”, the proposed K-CNN model is able to capture such linguistic clues and identify the causal relationship effectively. However, there are many complicated causal relations that are implicitly expressed, for example:

1. *The car ⟨e_1⟩accident⟨/e_1⟩ **made** she ⟨e_2⟩disabled⟨/e_2⟩.*
2. *The ⟨e_1⟩company⟨/e_1⟩ **made** a new ⟨e_2⟩product⟨/e_2⟩.*

There is causal relationship between $e1$ and $e2$ in sentence 1, however there is no causal relationship in sentence 2. People understand such causal relationship by performing mental operations based on their experiences or common senses, such knowledge may not be reflected on text itself, even though word embeddings are able to capture semantic meanings of words, the model requires large amount of data to learn. Therefore, more semantic features from external knowledge resources are needed to identify complicated causal relations. In this paper, we propose two types of semantic features.

3.3.1. Wordnet categorical features

Each word sense in WordNet hierarchy has a top level category, for example, there are 26 categories for nouns (such as location, event, state) and 15 categories for verbs (such as change, communication, emotion). Some categories such as “change”, “event”, “state” etc. have higher possibilities to be involved in a causal relationship, and some categories such as “animal”, “plant”, “communication” etc. have lower possibilities. Therefore, we find the WordNet top level categories of the two target entities (e_1 and e_2) and incorporate these categorical information into our model.

In the above example, the WordNet category of “accident” in sentence 1 is “event”, and the WordNet category of “company” in sentence 2 is “group”. By learning from data, the model is able to conclude that “event” category has much higher probability to be involved in causal relationship than “group” category, hence distinguish the two sentences. As majority of the target entities involved in causal relations are nouns and verbs, we only consider noun categories and verb categories of WordNet. The categorical information of e_1 and e_2 are encoded as categorical features using one-hot encoding, and these categorical features are concatenated with the feature vector from K-CNN and then passed to the classifier.

3.3.2. Framenet causal scores

Based on the causal frames in FrameNet, we compute causal scores to reflect the extent to which a sentence involves causal relationship. Similar as word filter bank generation in Section 3.1.2, we find all the lexical units of causal frames which represent keywords and cue phrases of causal relationship. The difference is that we focus on the ambiguity of causal words, which means the word may indicate other semantic frames other than causal frames. For example, the word “make” invokes 10 semantic frames besides causal frame, such as “Building”, “Cooking_creation” and “Manufacturing” frames. Such ambiguity is captured by calculating the probability of the lexical unit w_i invoking causal frames based on the 170,000 annotated sentence exemplars in FrameNet corpus, as shown in Eq. (6).

$$p(c \mid w_i) = \begin{cases} \|\text{sent}_{cf}\| / \|\text{sent}\| & \text{if } w_i \in \text{causal frames} \\ 0 & \text{if } w_i \notin \text{causal frames} \end{cases} \tag{6}$$

where $\|\text{sent}_{cf}\|$ indicates number of causal frame exemplars invoked by w_i, $\|\text{sent}\|$ indicates total number of exemplars invoked by w_i.

The probability $p(c|w_i)$ is used as the causal score of the lexical unit w_i, and the causal score for a sentence is calculated by adding the causal score of each lexical unit in the sentence.

$$s = \sum_i p(c \mid w_i) \tag{7}$$

Besides the causal score for the whole sentence, we also compute three additional causal scores for three different locations in the sentence respectively, including words before e_1 (including e_1), words after e_2 (including e_2) and words between e_1 and e_2. The FrameNet causal score features are also concatenated with the feature vector from K-CNN and then passed to the classifier.

3.4. Regularization and classification

The final feature vector $\mathbf{p} \in \mathbb{R}^{h+r+a}$ consists of the knowledge-oriented channel output, data-oriented channel output and additional features, which represents the high-level features of the input sentence extracted by our model. Before passing the feature vector to the classifier to make the final judgment of causality, we apply dropout regularization (Hinton, Srivastava, Krizhevsky, Sutskever, & Salakhutdinov, 2012) to prevent co-adaptation of hidden units by randomly setting a proportion ρ of the feature vector to zero, as shown in Eq. (8).

$$\mathbf{p}_d = \mathbf{p} \circ \mathbf{b} \tag{8}$$

where $\circ$ represents element-wise multiplication and $\mathbf{b} \in \mathbb{R}^{r+h+a}$ is a vector of Bernoulli random variables with probability ρ of being 0. The feature vector after dropout $\mathbf{p}_d$[3] is then fed into a classifier to predict the class label. The classifier consists of a fully connected layer of standard neural network and a softmax layer to predict the class probabilities.

We train the model by minimizing the categorical cross entropy loss function. The free parameters for training include the position embedding matrix, data-oriented channel filter weights, and classifier weights including fully connected layer and softmax layer weights. Training is done using mini-batch stochastic gradient descent (SGD) with the Adadelta update rule (Zeiler, 2012).

4. Experiments

4.1. Datasets

Three datasets are used to evaluate our model, which are generated from SemEval-2010 task 8 dataset (Hendrickx et al., 2009), Causal-TimeBank dataset (Mirza, Sprugnoli, Tonelli, & Speranza, 2014) and Event StoryLine dataset (Caselli & Vossen, 2017).

SemEval-2010 task 8 dataset[4] contains 10,717 annotated samples, each sample is a sentence annotated with a pair of entities (e_1 and e_2) and their relationship class. Besides *Cause-Effect* relation which is directed[5], there are additional 8 directed relation classes and one undirected *Other* class. As we only interested in *Cause-Effect* relation, we annotate all other relations as *Other* class, resulting 478 *Cause-Effect(e_1, e_2)* samples, 853 *Cause-Effect(e_2, e_1)* samples and 9386 *Other* samples.

Causal-TimeBank dataset[6] is a causally annotated dataset based on TempEval-3 TimeBank data. The causal relations and their directions are marked with C-SIGNALs (causal signals) and CLINKs (causal links) tags, we extract all the causal relations between two entities and annotate them in the same way as SemEval-2010 task 8. There are totally 137 *Cause-Effect(e_1, e_2)* samples and 161 *Cause-Effect(e_2, e_1)* samples generated from Causal-TimeBank dataset, and we randomly select 300 annotated entity pairs which are not causally related as *Other* class.

Event StoryLine dataset[7] is a new benchmark dataset for temporal and causal relation detection, the corpus for annotation contains 258 documents concerning calamity events. We extract all the samples with causal relations by looking into the “CAUSES” and “CAUSED_BY” attributes in the “PLOT_LINK” tag and annotate them in the same format as SemEval-2010 task 8 dataset. There are totally 67 *Cause-Effect(e_1, e_2)* samples and 45 *Cause-Effect(e_2, e_1)* samples generated from Event StoryLine dataset, and we randomly select 220 sample sentences with no causal relationship as *Other* class.

The dataset generated from SemEval-2010 task 8 contains a large amount of samples, majority of causal relations in this dataset are simple causal relations which are explicitly expressed using linguistic clues such as “cause”, “result in”, “trigger”, “produce” etc. Besides, the data is unbalanced and more biased toward the *Other* class, hence it is suitable for evaluating explicit causal relation extraction. The data generated from Causal-TimeBank and Event StoryLine datasets contain more complex causal relations with no specific linguistic clues, which are complicated and hard to identify. Besides, the two datasets are much smaller than SemEval-2010 task 8 dataset, hence model overfitting issue is much more severe on these two datasets.

4.2. Experiment settings

To investigate the performance of our proposed K-CNN model, we chose two conventional CNN models as competitors to make

[3] Dropout is only applied during training time, feature vector $\mathbf{p}$ (without dropout) is used at test time.

[4] The dataset can be downloaded from https://docs.google.com/document/d/1QO_CnmvNRnYwNWu1-QCAeR5ToQYkXUqFeAJbdEhsq7w/preview.

[5] *Cause-Effect(e_1, e_2)* means e_1 causes e_2, *Cause-Effect(e_2, e_1)* means e_2 causes e_1.

[6] The dataset can be requested for download on https://hlt-nlp.fbk.eu/technologies/causal-timebank.

[7] The dataset can be downloaded from https://github.com/cltl/EventStoryLine.git.

comparison, one is proposed by Zeng et al. (2014) which uses filters with single window size ($k = 3$), we name it *CNN_Single*; the other is proposed by Nguyen and Grishman (2015) which uses filters with multiple window sizes ($k = 2, 3, 4, 5$), we name it *CNN_multiple*.

To make fair comparison, all models including ours use Dependency-Based Word Embeddings[8] that is pre-trained using English Wikipedia, the word embeddings are unit vectors with a dimensionality of $e = 300$, and they are kept static during training; the dimensionality of position embeddings is $d = 20$; the dropout rate is $\rho = 0.4$; the mini-batch size for training is 20; the dimensionality of hidden layer before softmax layer is half of the feature vector dimension. For words not in the word embeddings, we initialize them as random unit vectors with same dimensionality as word embeddings. Besides the above mentioned hyper-parameters, the optimal number of filters for each model are found through grid search (as illustrated in Section 4.3.1 below). The evaluation metric is macro-averaged F1 score calculated from 10-fold cross-validation. To produce stable results, the 10-fold cross-validation is run 10 times for each experiment, and the data is shuffled with different random seeds before the cross-validation. We take the average value of the 10 macro-averaged F1 scores as the final result.

We used five variations of K-CNN model to study the effect of our proposed methods, including K-CNN, K-CNN_K, K-CNN_FS, K-CNN_FS+C_{max}, and K-CNN_FS+C_{avg}. K-CNN is our base model (without word filter selection and clustering), K-CNN_K is the K-CNN model using knowledge-oriented channel only, FS means word filter selection, C_{max} means clustering of word filters with max-pooling within each cluster, and C_{avg} means clustering of word filters with average-pooling within each cluster. For word filter selection process, the critical F-ratio is $F_\alpha = 2.9957$ by assuming the degree of freedom for the denominator is $+\infty$. For word filters clustering, we use half of the number of word filters as the number of clusters, hence reducing the dimensionality of the feature vector from knowledge-oriented channel by half.

4.3. Results and analyses

4.3.1. Effect of data-oriented channel

To study the effect of data-oriented channel and find its optimal number of filters, we perform grid searching of the number of filters ranging from 0 to 100. We use the macro-averaged F1 score calculated from 10-fold cross-validation for evaluation. The grid search results for the three datasets are plotted in Fig. 3.

From Fig. 3, we can find the optimal number of filters for each window size in data-oriented channel is between 25 to 30. Further increasing the number of filters will cause the drop of F1 score and the model performance is not stable. This is because that the model is prone to overfitting when the number of free parameters is large and the training data is limited. Compared with conventional CNNs which normally use hundreds of convolutional filters, the number of filters needed in the data-oriented channel of K-CNN is significantly reduced.

When the number of filters of data-oriented channel is 0, the model only uses knowledge-oriented channel of K-CNN (K-CNN_K). Table 1 shows a comparison of K-CNN_K and K-CNN with optimal number of filters in data-oriented channel. Experiment result shows that the data-oriented channel can improve both precision and recall of the model. The reason is that data-oriented channel is able to extract important features which are neglected by knowledge-oriented channel by learning from data. It involves the whole sentence and able to capture longer dependencies by using convolutional filters with longer window sizes. Therefore, it is necessary to have data-oriented channel in K-CNN which plays a supplementary role to knowledge-oriented channel and gives the model enough capacity to learn important information from data.

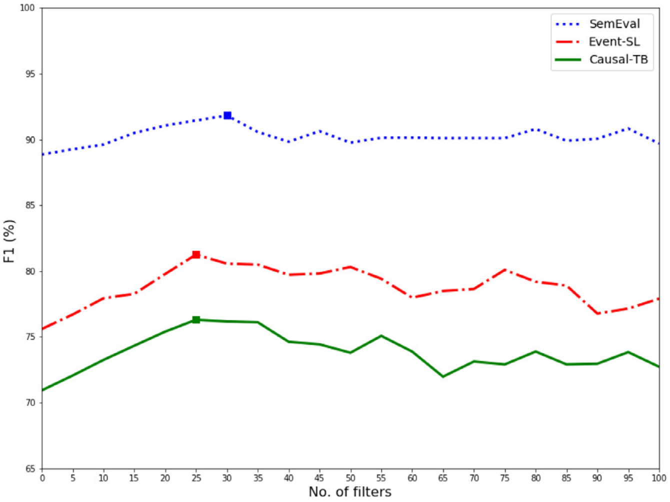

Fig. 3. Effect of the number of filters in data-oriented channel. “SemEval”, “Event-SL”, “Causal-TB” indicate SemEval-2010 task 8, Event StoryLine and Causal-TimeBank datasets, respectively.

Table 1
Macro-averaged precision (P), recall (R) and F1 scores of K-CNN without and with data-oriented channel. Scores are shown in percentage(%).

Datasets		K-CNN_K	K-CNN
SemEval	P	92.74	93.12
	R	85.56	90.73
	F1	88.85	**91.82**
Causal-TB	P	71.84	76.91
	R	70.24	75.85
	F1	70.92	**76.29**
Event-SL	P	77.38	82.07
	R	74.11	80.58
	F1	75.60	**81.25**

Table 2
Macro-averaged F1 score (%) of various model architectures for causal relation extraction task on three datasets.

Models	SemEval	Causal-TB	Event-SL
CNN_single	90.15	72.67	75.72
CNN_multiple	90.84	73.37	76.45
K-CNN	91.49	75.83	80.64
K-CNN_FS	91.90	75.96	81.27
K-CNN_FS+Cmax	**92.34**	**76.21**	81.84
K-CNN_FS+Cavg	91.20	75.43	**81.96**

4.3.2. Effect of knowledge-oriented channel and dimensionality reduction

Table 2 shows the macro-averaged F1 scores of the above mentioned models for causal relation extraction on the three datasets. For conventional CNN models, we found that CNN_multiple performs slightly better than CNN_single, the reason is that CNN_multiple uses multiple convolutional window sizes, which is able to capture more information of causality from different n-grams. Comparing our proposed K-CNN models with conventional CNN models, the improvements of the experiment results demonstrate that K-CNN models with knowledge-oriented word filters are able to extract causal relations more effectively. We attribute the performance advantage of K-CNN over conventional CNN models to the following three facts:

[8] The pre-trained word embedding is available at https://levyomer.wordpress.com/2014/04/25/dependency-based-word-embeddings.

- The word filters constructed from lexical knowledge bases have more precise representation and broader coverage of the keywords and cue phrases of causal relationship than random initialized convolutional filters, which allow the model to extract the linguistic clues of causal relation from sentence more effectively.
- The combination of knowledge-oriented channel and data-oriented channel enables the model to capture significant features of causal relationship. The knowledge-oriented channel takes advantage of existing knowledge bases and captures significant linguistic clues of causal relationship; the data-oriented channel learns other important features such as features with long distance dependencies from the data. The two channels complement with each other, resulting a good model for causal relation extraction.
- As the word filters are static during training, majority of free parameters of K-CNN lie in data-oriented channel, which are the weights of convolutional filters. The number of free parameters in data-oriented channel depends on the number and the dimensionality of convolutional filters, which is equal to $(e+2d)\times(3r+4r)=(300+2\times20)\times7\times25=59,500$, where r is the optimal number of filters for each window size, which is found to be 25 as shown in Section 4.3.1 (we ignore the bias term). Whereas for CNN_single and CNN_multiple, the numbers of free parameters are $(e+2d)\times3r=163,200$ and $(e+2d)\times(2r+3r+4r+5r)=238,000$ respectively, where r is 160 for CNN_single and 50 for CNN_multiple, which are found by grid search. Therefore, the number of free parameters of K-CNN is reduced significantly compared with conventional CNN models. This alleviates overfitting issue when the amount of training data is small, such as the datasets generated from Causal TimeBank and Event StoryLine. That's why the performance improvements for the two small datasets are more significant than SemEval-2010 task 8 dataset which is relatively larger.

To further investigate the effectiveness of K-CNN in alleviating overfitting, for each fold in the 10-fold cross-validation, we train K-CNN as well as conventional CNN (we experiment on CNN_multiple) on the training set and test the models on both training set and validation set. Then we plot the learning curves which show the macro-averaged F1 scores on training set and validation set with increasing number of training epochs. We observed similar trends of the learning curves for all the 10 folds, hence only one of the 10 folds is shown in this paper (Fig. 4). It is observed that K-CNN consistently outperforms conventional CNN during training and testing stage. With increasing number of training epochs, the scores on training sets can eventually reach 100% for both models, however, the scores on validation sets cannot be further improved after certain training epochs. The gap between training scores and validation scores is actually due to overfitting of the models on training data. As shown in Fig. 4, K-CNN achieves higher scores on validation sets compared with conventional CNN, this demonstrates the ability of K-CNN in alleviating overfitting issue. Besides, training is done much faster for K-CNN (within few epochs) compared with conventional CNN, and the difference of training efficiency is significant for the first epoch. The reason is that conventional CNN needs to train all the parameters from scratch, whereas only the data-oriented channel in K-CNN needs to be trained and the knowledge-oriented channel has already contained many useful information for extracting causal relationship before training.

Furthermore, the experiment results in Table 2 also show that the proposed word filter selection and clustering for dimensionality reduction can improve the performance of K-CNN. K-CNN with word filter selection (K-CNN_FS) using ANOVA F-ratio hypothesis testing performs better than K-CNN, this is because that the non-discriminative features produced by the insignificant word filters are considered as noises and will hinder the classification accuracy. K-CNN_FS removes such word filters and only keeps the word filters which produce discriminative features with large separability among class means. Further clustering of the features based on the similarities of word filters also improves the performance of K-CNN, as the redundant features produced by semantically similar word filters are removed and the feature dimensionality is reduced, resulting a more compact and higher quality feature vector. The experiment results show that max-pooling within clusters (C_{max}) generally outperforms average-pooling (C_{avg}), and the classi-

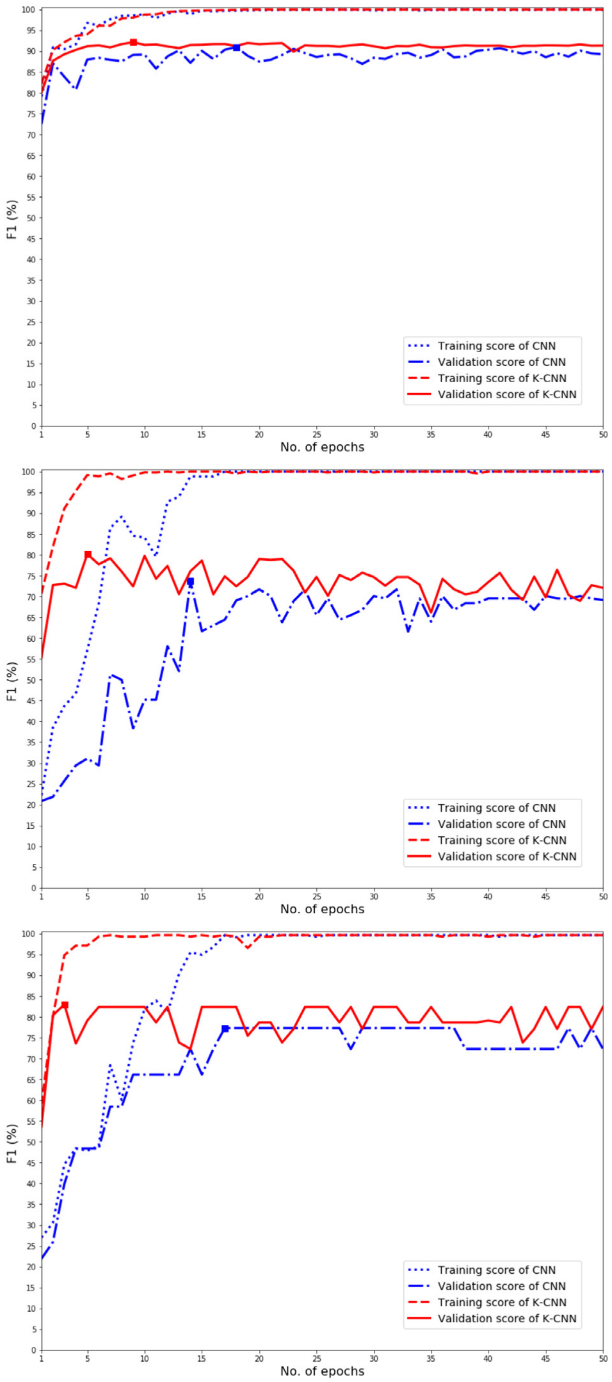

Fig. 4. Learning curves of K-CNN and CNN_multiple for SemEval-2010 task 8 (top), Causal-TimeBank (middle) and Event StoryLine (bottom) datasets. Squared dot indicates the highest score achieved on validation set.

Table 3
Macro-averaged precision (P), recall (R) and F1 scores of K-CNN without and with semantic features (SF). Scores are shown in percentage (%).

Datasets		K-CNN w/o SF	K-CNN with SF
SemEval	P	93.37	94.61
	R	91.40	90.87
	F1	92.34	**92.64**
Causal-TB	P	76.94	78.71
	R	75.70	77.26
	F1	76.21	**77.86**
Event-SL	P	82.78	84.57
	R	81.03	82.15
	F1	81.84	**83.31**

fication results of max-pooling are more stable. The reason is probably that the significant clues of causality are preserved by max-pooling, however they are affected by the outliers in the cluster for average-pooling.

4.3.3. *Effect of additional semantic features*

We add the semantic features proposed in Section 3.3 to the best K-CNN model, K-CNN_FS+$\mathrm{C}_{\mathbf{max}}$, then the macro-averaged precision, recall and F1 score are evaluated. Table 3 shows the comparison results before and after adding the semantic features. Results show that the proposed semantic features are able to improve the F1 score for causal relation extraction. Furthermore, the performance improvements for Causal-TimeBank and Event StoryLine datasets are more significant than SemEval-2010 task 8 dataset, indicating that the proposed semantic features are especially useful for complex and implicitly expressed causal relations, as there are more such causal relations in Causal-TimeBank and Event StoryLine datasets compared with SemEval-2010 task 8 dataset. We observed that both precision and recall are improved for the two datasets by adding the semantic features. We attribute the performance improvements to the ability of the proposed semantic features for finding the useful semantic information hiding behind the text for causal relation extraction.

5. Conclusion and future work

In this work, we propose a Knowledge-oriented Convolutional Neural Network (K-CNN) for causal relation extraction. We present an effective way of combing human prior knowledge from lexical knowledge bases and information from data for CNN to achieve better performance. The proposed K-CNN contains two convolutional channels: knowledge-oriented channel and data-oriented channel. The convolutional filters (word filters) of knowledge-oriented channel are automatically generated based on the linguistic knowledge of causal relationship in lexical knowledge bases including WordNet and FrameNet. These word filters represent significant linguistic clues of causal relationship, allowing the model to extract these linguistic clues precisely and effectively. The convolutional filters in data-oriented channel are trained based on the training data, and the channel uses position embeddings and wider convolutional window size to capture longer dependencies and other useful features from the whole sentence. The proposed K-CNN is fully automatic without sophisticated feature engineering. In addition, the model addresses the overfitting issue caused by the lack of training data by reducing the number of free parameters of the model, yet achieves better performance for causal relation extraction compared with conventional CNNs.

Word filter selection and clustering techniques are proposed to reduce dimensionality and further improve the performance of K-CNN model. We also propose additional semantic features to address more complex causal relations. The performance of our approaches has been experimentally verified using three datasets for causal relation extraction. In future work, the automatic selection of the target entities for causal relation identification and the more effective extraction of complex causal relations are to be investigated. Furthermore, we will try to apply K-CNN to other relation extraction tasks besides causal relation, and explore potential applications of K-CNN in other domain-specific tasks.

References

Ackerman, E. J. M. (2012). Extracting a causal network of news topics. In *Proceedings of the OTM confederated international conferences "on the move to meaningful internet systems"* (pp. 33–42). Springer.

Araúz, P. L., & Faber, P. (2012). Causality in the specialized domain of the environment. In *Proceedings of the semantic relations-II. Enhancing resources and applications workshop programme.* (p. 10). Citeseer.

Bethard, S., & Martin, J. H. (2008). Learning semantic links from a corpus of parallel temporal and causal relations. In *Proceedings of the forty sixth annual meeting of the association for computational linguistics on human language technologies: Short papers* (pp. 177–180). Association for Computational Linguistics.

Blanco, E., Castell, N., & Moldovan, D. I. (2008). Causal relation extraction.. In *Proceedings of the language resources and evaluation conference, LREC.*

Caselli, T., & Vossen, P. (2017). The event storyline corpus: A new benchmark for causal and temporal relation extraction. In *Proceedings of the events and stories in the news workshop* (pp. 77–86).

Chan, K., & Lam, W. (2005). Extracting causation knowledge from natural language texts. *International Journal of Intelligent Systems, 20*(3), 327–358.

Chang, D.-S., & Choi, K.-S. (2004). Causal relation extraction using cue phrase and lexical pair probabilities. In *Proceedings of the International conference on natural language processing* (pp. 61–70). Springer.

Cheong, H., & Shu, L. (2012). Automatic extraction of causally related functions from natural-language text for biomimetic design. In *Proceedings of the international design engineering technical conferences and computers and information in engineering conference* (pp. 373–382). American Society of Mechanical Engineers.

David, M. L. (2018). Online statistics education: An interactive multimedia course of study. http://onlinestatbook.com.

Fellbaum, C. (2010). Theory and applications of ontology: Computer applications. Wordnet.

Girju, R. (2003). Automatic detection of causal relations for question answering. In *Proceedings of the ACL workshop on multilingual summarization and question answering-volume 12* (pp. 76–83). Association for Computational Linguistics.

Girju, R., Beamer, B., Rozovskaya, A., Fister, A., & Bhat, S. (2010). A knowledge-rich approach to identifying semantic relations between nominals. *Information Processing & Management, 46*(5), 589–610.

Girju, R., Moldovan, D. I., et al. (2002). Text mining for causal relations.. In *Proceedings of the flairs conference* (pp. 360–364).

Goldvarg, E., & Johnson-Laird, P. N. (2001). Naive causality: A mental model theory of causal meaning and reasoning1. *Cognitive Science, 25*(4), 565–610.

Gómez-Adorno, H., Posadas-Durán, J.-P., Sidorov, G., & Pinto, D. (2018). Document embeddings learned on various types of n-grams for cross-topic authorship attribution. *Computing, 100*(7), 741–756.

Goodfellow, I., Bengio, Y., Courville, A., & Bengio, Y. (2016). *Deep learning*: 1. MIT press Cambridge.

Hashimoto, C., Torisawa, K., Kloetzer, J., Sano, M., Varga, I., Oh, J.-H., & Kidawara, Y. (2014). Toward future scenario generation: Extracting event causality exploiting semantic relation, context, and association features. In *Proceedings of the fifty second annual meeting of the association for computational linguistics (volume 1: Long papers): 1* (pp. 987–997).

Hashimoto, K., Miwa, M., Tsuruoka, Y., & Chikayama, T. (2013). Simple customization of recursive neural networks for semantic relation classification. In *Proceedings of the conference on empirical methods in natural language processing* (pp. 1372–1376).

Hendrickx, I., Kim, S. N., Kozareva, Z., Nakov, P., Ó Séaghdha, D., Padó, S., et al. (2009). Semeval-2010 task 8: Multi-way classification of semantic relations between pairs of nominals. In *Proceedings of the workshop on semantic evaluations: Recent achievements and future directions* (pp. 94–99). Association for Computational Linguistics.

Higashinaka, R., & Isozaki, H. (2008). Automatically acquiring causal expression patterns from relation-annotated corpora to improve question answering for why-questions. *ACM Transactions on Asian Language Information Processing (TALIP), 7*(2), 6.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2012). Improving neural networks by preventing co-adaptation of feature detectors. Computing Research Repository (CoRR), arXiv:1207.0580.

Hovy, E., Marcus, M., Palmer, M., Ramshaw, L., & Weischedel, R. (2006). Ontonotes: the 90% solution. In *Proceedings of the human language technology conference of the NAACL, companion volume: Short papers* (pp. 57–60). Association for Computational Linguistics.

Inui, T., Inui, K., & Matsumoto, Y. (2005). Acquiring causal knowledge from text using the connective marker tame. *ACM Transactions on Asian Language Information Processing (TALIP), 4*(4), 435–474.

Ittoo, A., & Bouma, G. (2011). Extracting explicit and implicit causal relations from sparse, domain-specific texts. In *Proceedings of the International conference on application of natural language to information systems* (pp. 52–63). Springer.

Jensen, L. J., Saric, J., & Bork, P. (2006). Literature mining for the biologist: From information retrieval to biological discovery. *Nature Reviews Genetics, 7*(2), 119.

Kalchbrenner, N., Grefenstette, E., & Blunsom, P. (2014). A Convolutional Neural Network for Modelling Sentences. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)* (pp. 655–665). Baltimore, Maryland: Association for Computational Linguistics.

Khoo, C. S., Kornfilt, J., Oddy, R. N., & Myaeng, S. H. (1998). Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing. *Literary and Linguistic Computing, 13*(4), 177–186.

Kim, J.-D., Ohta, T., & Tsujii, J. (2008). Corpus annotation for mining biomedical events from literature. *BMC Bioinformatics, 9*(1), 10.

Kim, Y. (2014). Convolutional neural networks for sentence classification. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 1746–1751).

Kingsbury, P., & Palmer, M. (2003). Propbank: The next level of treebank. In *Proceedings of the treebanks and lexical theories: 3*. Citeseer.

Le, Q., & Mikolov, T. (2014). Distributed representations of sentences and documents. In *Proceedings of the international conference on machine learning* (pp. 1188–1196).

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE, 86*(11), 2278–2324.

Lee, J. Y., Dernoncourt, F., & Szolovits, P. (2017). Mit at semeval-2017 task 10: Relation extraction with convolutional neural networks. In *Proceedings of the 11th International Workshop on Semantic Evaluations (SemEval-2017)* (pp. 978–984). Association for Computational Linguistics.

Lee, K. C., & Lee, S. (2012). A causal knowledge-based expert system for planning an internet-based stock trading system. *Expert Systems with Applications, 39*(10), 8626–8635.

Mele, F., & Sorgente, A. (2013). Ontotimefl–a formalism for temporal annotation and reasoning for natural language text. In *Proceedings of the new challenges in distributed information filtering and retrieval* (pp. 151–170). Springer.

Mikolov, T., Corrado, G., Chen, K., & Dean, J. (2013). Efficient Estimation of Word Representations in Vector Space. In *Proceedings of the International Conference on Learning Representations (ICLR 2013)* (pp. 1–12).

Mirza, P., Sprugnoli, R., Tonelli, S., & Speranza, M. (2014). Annotating causality in the tempeval-3 corpus. In *Proceedings of the EACL workshop on computational approaches to causality in language (CAtoCL)* (pp. 10–19).

Nguyen, T. H., & Grishman, R. (2015). Relation extraction: Perspective from convolutional neural networks. In *Proceedings of the first workshop on vector space modeling for natural language processing* (pp. 39–48).

Pakray, P., & Gelbukh, A. (2014). An open-domain cause-effect relation detection from paired nominals. In *Proceedings of the Mexican international conference on artificial intelligence* (pp. 263–271). Springer.

Peña, A., Sossa, H., & Gutiérrez, A. (2008). Causal knowledge and reasoning by cognitive maps: Pursuing a holistic approach. *Expert Systems with Applications, 35*(1–2), 2–18.

Posadas-Durán, J.-P., Gómez-Adorno, H., Sidorov, G., Batyrshin, I., Pinto, D., & Chanona-Hernández, L. (2017). Application of the distributed document representation in the authorship attribution task for small corpora. *Soft Computing, 21*(3), 627–639.

Radinsky, K., Davidovich, S., & Markovitch, S. (2012). Learning causality for news events prediction. In *Proceedings of the twenty first international conference on world wide web* (pp. 909–918). ACM.

Riaz, M., & Girju, R. (2014). Recognizing causality in verb-noun pairs via noun and verb semantics. In *Proceedings of the EACL workshop on computational approaches to causality in language (CAtoCL)* (pp. 48–57).

Rink, B., Bejan, C. A., & Harabagiu, S. M. (2010). Learning textual graph patterns to detect causal event relations.. In *Proceedings of the flairs conference*.

Rink, B., & Harabagiu, S. (2010). UTD: Classifying semantic relations by combining lexical and semantic resources. In *Proceedings of the fifth international workshop on semantic evaluation* (pp. 256–259). Association for Computational Linguistics.

Ruppenhofer, J., Ellsworth, M., Petruck, M. R., Johnson, C. R., & Scheffczyk, J. (2016). *FrameNet II: Extended theory and practice*. Institut für Deutsche Sprache, Bibliothek.

Santos, C. N. D., Xiang, B., & Zhou, B. (2015). Classifying relations by ranking with convolutional neural networks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)* (pp. 626–634).

Schuler, K. K. (2005). VerbNet: A broad-coverage, comprehensive verb lexicon. *Ph.D. thesis University of Pennsylvania*.

Socher, R., Huval, B., Manning, C. D., & Ng, A. Y. (2012). Semantic compositionality through recursive matrix-vector spaces. In *Proceedings of the joint conference on empirical methods in natural language processing and computational natural language learning* (pp. 1201–1211). Association for Computational Linguistics.

Sorgente, A., Vettigli, G., & Mele, F. (2013). Automatic extraction of cause-effect relations in natural language text. In *CEUR Workshop Proceedings: Vol. 1109* (pp. 37–48). CEUR-WS.

Turian, J., Ratinov, L., & Bengio, Y. (2010). Word representations: a simple and general method for semi-supervised learning. In *Proceedings of the forty eighth annual meeting of the association for computational linguistics* (pp. 384–394). Association for Computational Linguistics.

Wang, S., Zhe, Z., Kang, Y., Wang, H., & Chen, X. (2008). An ontology for causal relationships between news and financial instruments. *Expert Systems with Applications, 35*(3), 569–580.

Yang, X., & Mao, K. (2014). Multi level causal relation identification using extended features. *Expert Systems with Applications, 41*(16), 7171–7181.

Zeiler, M. D. (2012). Adadelta: an adaptive learning rate method. Computing Research Repository (CoRR), abs/1212.5,6. Retrieved from 1212.5701.

Zeng, D., Liu, K., Lai, S., Zhou, G., & Zhao, J. (2014). Relation classification via convolutional deep neural network. In *Proceedings of the COLING twenty fifth international conference on computational linguistics: Technical papers* (pp. 2335–2344).

Zhao, S., Liu, T., Zhao, S., Chen, Y., & Nie, J.-Y. (2016). Event causality extraction based on connectives analysis. *Neurocomputing, 173*, 1943–1950.