Discovering Causal Relations in Textual Instructions

Kristina Yordanova
University of Rostock
kristina.yordanova@uni-rostock.de

Abstract

One aspect of ontology learning methods is the discovery of relations in textual data. One kind of such relations are causal relations. Our aim is to discover causations described in texts such as recipes and manuals. There is a lot of research on causal relations discovery that is based on grammatical patterns. These patterns are, however, rarely discovered in textual instructions (such as recipes) with short and simple sentence structure. Therefore we propose an approach that makes use of time series to discover causal relations. We distinguish causal relations from correlation by assuming that one word causes another only if it precedes the second word temporally. To test the approach, we compared the discovered by our approach causal relations to those obtained through grammatical patterns in 20 textual instructions. The results showed that our approach has an average recall of 41% compared to 13% obtained with the grammatical patterns. Furthermore the discovered by the two approaches causal relations are usually disjoint. This indicates that the approach can be combined with grammatical patterns in order to increase the number of causal relations discovered in textual instructions.

1 Introduction and Motivation

There is an increasing number of approaches and systems for ontology learning based on textual data partially because of the availability of web resources that are easily accessible on the internet (Wong et al., 2012). One problem these approaches face is the discovery of relations in the data (Wong et al., 2012). One type of relations are the causal relations between text elements, that is, whether one word or phrase causes another. Most of the research regarding causal relations is centred on the discovery of causal relations between topics (Radinsky et al., 2011; Kim et al., 2012; Li et al., 2010) or based on a large amount of textual data (Silverstein et al., 2000; Mani and Cooper, 2000; Girju, 2003). Moreover, the works usually focus on the discovery of causal relations in rich textual data with complex sentence structure (Silverstein et al., 2000; Mani and Cooper, 2000; Girju, 2003). There is however little research on discovering causal relations in textual instructions that have short sentence length and simple structure (Zhang et al., 2012). This can be explained with the fact that short sentences often do not contain any grammatical causal patterns, rather the relations are implicitly inferred by the reader. There is a large amount of web instructions available in the form of recipes, manuals, and tutorials[1] that contain such simple structures. For example, in the sentence *"Add the pork pieces, fry them for 2 minutes."* there is no explicit causal relation between *add* and *fry*. However, we implicitly know, that without adding the pork pieces, we cannot fry them. This means that when attempting to learn an ontology representing the domain knowledge of such domain, it is difficult to discover causal relations between the ontology elements. For example, when attempting to learn the ontology structure of our experimental data with a state of the art tool (Cimiano and Völker, 2005), it is able to identify *is-a* relations, but no similarity or causal relations in the text. To address the problem of identifying causal relations in textual data, in this paper we discuss an approach that utilises time series in order to find temporally dependent elements in the text. We concentrate on the discov-

[1]For example, *BBC Food Recipes* provides currently 12 385 recipes (BBC, 2015).

Proceedings of Recent Advances in Natural Language Processing, pages 714–720,
Hissar, Bulgaria, Sep 7–9 2015.

ery of relations between events[2], and on the relation between events and the words that describe the changes these events cause.

The work is structured as follows. In Section 2 we discuss the related work on causal relations discovery. In Section 3 we present our approach to causality discovery. The experimental setup to test our approach is described in Section 4. Later, we discuss the results in Section 5 and we conclude the work with a discussion about the advantages and limitations of the approach (Section 6).

2 Related Work

There is a lot of research on discovery of causal relations in textual data. Most of it is centred on applying grammatical patterns in order to identify the relations. Khoo et al. (Khoo et al., 1998) propose five ways of explicitly identifying cause-effect pairs, and based on them construct patterns for discovering them. The patterns employ causal links between two phrases or clauses (e.g. *hence*, *therefore*), causative verbs (e.g. *cause*, *break*), resultative constructions (verb-noun-adjective constructions), conditionals (e.g. *if-then*), and causative adverbs and adjectives (e.g. *fatally*). Khoo et al. also provide an extensive catalogue of causative words and phrases. Based on this concept other works search for causal relations for different applications. For example, Li et al. attempt to generate attack plans based on newspaper data (Li et al., 2010); Girju et al. utilise grammatical patterns in order to analyse cause-effect questions in question answering system (Girju, 2003); Cole et al. apply grammatical patterns to textual data in order to obtain Bayesian network fragments (Cole et al., 2006); and Radinsky et al. mine web articles to identify causal relations (Radinsky et al., 2011).

Other approaches combine grammatical patterns with machine learning in order to extract preconditions and effects from textual data. For example, Sill et al. train a support vector machine with a large annotated textual corpus in order to be able to identify preconditions and effects, and to build STRIPS representations of actions and events (Sil and Yates, 2011).

Alternative approaches rely on the Markov condition to identify causal relations between documents. They utilise the LCD algorithm that tests variables for dependence, independence, and conditional independence to restrict the possible causal relations (Cooper, 1997). Based on this algorithm Silverstein et al. were able to discover causal relations between words by representing each article as a sample with the n most frequent words (Silverstein et al., 2000). Similarly, Mani et al. apply the LCD algorithm to identify causal relations in medical data (Mani and Cooper, 2000).

All of the above methods are applied to large amounts of data, usually with rich textual descriptions. There is, however, no much research on finding the causal relations within a textual instructions document, where the sentences are short and simple. Zahng et al. attempt to extract procedural knowledge from textual instructions (manuals and recipes) in order to build a procedural model of the instruction (Zhang et al., 2012). By applying grammatical patterns they are able to build a procedural model of each sentence. They, however, do not discuss the relations between the identified procedures, thus, do not identify any causal relations between the sentences.

In our work we identify implicit causal relations within and between sentences in a document. To do that we adapt the approach proposed by Kim et al. (Kim et al., 2012; Kim et al., 2013), where they search for causally related topics by representing each topic as a time series where each time stamp is represented by a document from the corresponding topic. In the following we explain how the approach can be adapted to identify causal relations within a textual document.

3 Discovering Causal Relations using Time Series

Textual instructions such as recipes and manuals have a simple sentence structure that does not contain many grammatical patterns, indicating explicit causal relations. On the other hand, we as humans are able to detect implicit relations, e.g. that one instruction can be executed only after another was already executed. In that case, we can either assume that the causal relation between events follows the temporal relation (i.e. each event causes the next), or we can attempt to identify only those events that are causally related. Similarly, to identify the effects one event has on the object, or the state of the object that allows the occurrence of the event, one can search for grammatical patterns. That will however only

[2]By *event* we mean the verb describing the action that has to be executed in an instruction.

identify relations within the sentence but not between sentences (unless they are connected with a causal link). For example, in the sentences *"Simmer (the sauce) until thickened. Add the pork, mix well for one minute."* using a grammatical pattern we will discover that *simmer* causes *thickened* (through the causal link *until*). However, it will not discover that the sauce has to *thicken*, in order to *add* the pork. A grammatical pattern will also not discover the relation between *add* and *mix*, as there is no causal link between them. To discover such implicit relations, we treat each word in textual instructions as a time series. Then we apply causality test on the pairs of words we are interested in to identify whether they are causally related or not.

We concentrate on three types of causal relations. These are discovering causal relation (1) between two events; (2) between an event and its effect on the state of object over which the event is executed; (3) between the state of the object before an event can be executed over it. By state of the object we mean the phrase that serves as an adjectival modifier or a nominal subject.

We consider a text to be a sequences of sentences divided by a sentence separator.

Definition 1 ***(Text)*** *A text I is a set of tuples $(S, C) = \{(s_1, c_1), (s_2, c_2), ..., (s_n, c_n)\}$ where S represents the sentence and C the sentence separator, with n being the length of the text.*

Each sentence in the text is then represented by a sequence of words, where each word has a tag describing its part of speech (POS) meaning.

Definition 2 ***(Sentence)*** *A sentence S is a set of tuples $(W, T) = \{(w_1, t_1), ..., (w_m, t_m)\}$ where W represents the words in the sentence, and T the corresponding POS tag assigned to the words. The sentence is m words long.*

In a text we have different types of words. We are most interested in verbs as they describe the events that cause other events or changes. More precisely, a verb $v \in W$ is a word where for the tuple (v, t) holds that $t =$ *verb*. We denote the set of verbs with V. The events are then verbs in their infinitive form or in present tense, as textual instructions are usually described in imperative form with a missing agent.

Definition 3 ***(Event)*** *An event $e \in V$ is a verb where for the tuple (e, t) holds that $t =$ verb_infinitive OR verb_present. For short we say $t =$ event.*

We are also interested in those nouns that are the direct (accusative) objects of the verb. A noun $n \in W$ is a word where for the tuple (n, t) holds that $t =$ *noun*. We denote the set of nouns with N. Then we define the object in the following manner.

Definition 4 ***(Object)*** *An object $o \in N$ of a verb v is the accusative object of v. We denote the relation between o and v as dobj(v, o), and any direct object-verb in a sentence s_n as a tuple $(v, o)_n$.*

We define the state of an object as the adjectival modifier or the nominal subject of an object.

Definition 5 ***(State)*** *A state $c \in W$ of an object o is a word that has one of the following relations with the object: amod(c, o), denoting the adjectival modifier or nsubj(c, o), denoting the nominal subject. We denote such tuple as $(c, o)_n$, where n is the sentence number.*

As in textual instructions the object is often omitted (e.g. *"Simmer (the sauce) until thickened."*), we also investigate the relation between an event and past tense verbs or adjectives that do not belong to an adjectival modifier or to nominal subject, but that might still describe this relation.

3.1 Generating time series

Given the definitions above, we can now describe each unique word in a text as a time series. Each element in the series is a tuple consisting of the number of the sentence in the text, and the number of occurrences of the word in the sentence.

Definition 6 ***(Time series)*** *A time series of a word w is a sequence of tuples $(D, F)_w = \{(1, f_1)_w, (2, f_2)_w, ..., (n, f_n)_w\}$ where $D = \{1, ..., n\}$ is the timestamp, and F is the number of occurrences of a word at the given timestamp. Here n corresponds to the sentence number in the text.*

Algorithm 1 Generate time series for a given object and the events applied on it.

```
Require: (V, O)                                        ▷ all event-object pairs in I
Require: m ∈ O                                         ▷ a unique object
for S_n in I do                                        ▷ for each sentence in a text
    V_n ← [w | t == event, (w, t) ← S_n]               ▷ extract the events
end for
U ← unique(V)                                          ▷ returns all unique events in I
N ← [unique(o) | (v, o) ← (V, O)]                      ▷ collect the unique objects in I
for u in U do                                          ▷ for each unique event in I
    i ← 1
    while i ≤ length(I) do
        for (v, o) in (V, O)_i do                      ▷ for each event-object pair in S_i
            (D, F)_{u,i} ← (i, count((v == u, o == m)))   ▷ calculate the number of occurrences of (u, m) for each sentence
            i ← i + 1
        end for
    end while
end for
return (D, F)_m                                        ▷ return the time series for all events w.r.t. an object
```

Generally, we can generate a time series for each kind of word in the corpus, as well as for each tuple of words. Here we concentrate on those describing or causing change in a state. That means we generate time series for all events and for all states that change an object. To generate time series for the events we distinguish two cases. The first is of events that are applied to objects (e.g. *"simmer the sauce"*). In that case, for each unique object o in the corpus we generate a time series that describes how often this object had a direct object relation with a verb v, namely we are looking for the number of occurrences of $(v, o)_n$ in each sentence s_n (see Algorithm 1).

Apart from the events that are applied to an object, there are such that do not have a direct object relation, or where the relation is not explicitly described (e.g. *"Mix (the pork) well for one minute."*). In that case, we also search for causal relations in events without considering their direct objects (see Algorithm 2).

Algorithm 2 Generate time series representing the events in a textual corpus

Require: U ▷ all unique events in I
Require: V_n ▷ all unique events in each sentence S_n
for u in U **do** ▷ for each unique event in I
    $i \leftarrow 1$
    while $i \leq length(I)$ **do**
        for v in V_i **do** ▷ for each event in S_i
            $(D, F)_{u,i} \leftarrow (i, count(v == u))$ ▷ calculate the number of occurrences of u for S_i
            $i \leftarrow i + 1$
        end for
    end while
end for
return (D, F) ▷ return the time series for all events

To investigate the causal relation between a state of the object and an event, we also generate time series describing the state. This is done by following the procedure described in Algorithm 1 where the (O, V) pair is replaced with (C, O) pair, and where we no longer extract events but rather states c. In order to include all states where the object is omitted, we also generate time series for each adjective or verb in past tense that could potentially describe a state. To do that we follow the procedure in Algorithm 2, where instead of events we search for adjectives or past tense verbs.

3.2 Searching for causality

In order to discover causal relations based on the generated time series, we make use of the Granger causality test. It is a statistical test for determining whether one time series is useful for forecasting another. More precisely, Granger testing performs statistical significance test for one time series, "causing" the other time series with different time lags using auto-regression (Granger, 1969). The causality relationship is based on two principles. The first is that the cause happens prior to the effect, while the second states that the cause has a unique information about the future values of its effect (Granger, 2001). Based on these assumptions, given two sets of time series x_t and y_t, we can test whether x_t Granger causes y_t with a maximum p time lag. To do that, we estimate the regression $y_t = a_o + a_1 y_{t-1} + ... + a_p y_{t-p} + b_1 x_{t-1} + ... + b_p x_{t-p}$. An F-test is then used to determine whether the lagged x terms are significant.

Algorithm 3 Identify causal relation between two words

Require: (D, F) ▷ all time series describing words of interest in a corpus
Require: L ▷ the lag in the Granger causality test
Require: *Th* ▷ significance threshold
Require: $u \in W$ ▷ a word which causal relation w.r.t. the rest of the words is tested
for w in W, $w \neq u$ **do** ▷ for each unique time series
    $C_{u,w} \leftarrow granger.Causality(((D, F)_u, (D, F)_w), L)$ ▷ calculate the causality between w and u
    if $p.value(C_{u,w}) \leq Th$ **then** ▷ the relation is significant
        $R_{u,w} \leftarrow C_{u,w}$ ▷ u causes w
    end if
end for
return R_u ▷ return the list of words with which u is causally related

We use the Granger causality test to search for causal relations between the generated time series (see Algorithm 3). Generally, for each two time series of interest, we perform Granger test, and if the p value of the result is under the significance threshold, we conclude that the first time series causes the second, hence the first word causes the second. The Granger causality test can be applied only on stationary time series. Otherwise, they have to be converted into stationary time series before applying the test (e.g. by taking the difference of every two elements in the series).

4 Experimental Setup

To test our approach, we selected 20 different instructions: 10 recipes from BBC Food Recipes[3], 3 washing machine instructions[4], 3 coffee machine instructions[5], 3 kitchen experiment instructions describing the experiments from the CMU Grand challenge dataset[6], and one description of a cooking task experiment[7]. The shortest instruction is 5 lines (each line being a sentence with a

[3] `http://www.bbc.co.uk/food/recipes/`
[4] `http://www.miele.co.uk/Resources/OperatingInstructions/W%203923%20WPS.pdf`
[5] `http://www.cn.jura.com/service_support/download_manual_jura_impressa_e10_e20_e25_english.pdf`
[6] `http://kitchen.cs.cmu.edu/`
[7] Source not shown due to blind reviewing.

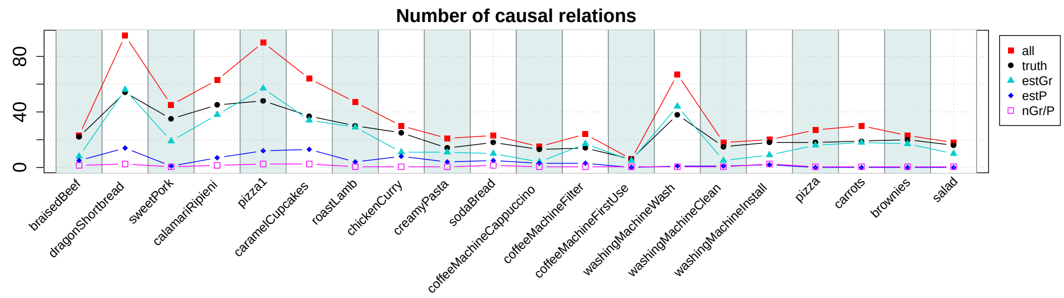

Figure 1: Number of causal relations discovered by a human expert (circle), Granger causality (triangle), part of speech patterns (rhombus), and all discovered relations (solid square). The square without fill shows the causal relations that have been discovered by both Granger causality and grammatical patterns.

full stop at the end), the longest is 111 lines, with a mean length of 31 lines. The average sentence length in an instruction text is 11.2 words, with the shortest text having an average of 5.7 words per sentence, and the longest an average of 17.4 words per sentence. The average number of events per sentence is 1.6, with the minimum average of 1 event per sentence in a text, and the average maximum of 2.23 events per sentence.

A human expert was asked to search for causal relations in the text, concentrating on relations between events or between states and events. This was later used as the ground truth against which the discovered relations were compared.

Later, each of the instructions was parsed by the Stanford NLP Parser[8] in order to obtain the part of speech tags and the dependencies between the words. This was then used as an input for generating the time series. We considered as a sentence separator a full stop and a comma, as in this type of instructions it divides sequentially executed events in one sentence. The time series were then tested for stationarity by using the Augmented Dickey–Fuller (ADF) t-statistic test. It showed that the series are already stationary.

We search for causal relations between events without considering the object, between events given the object, and between events and states. For the case of events given the object we performed Granger causality test with a lag from 1 to 5 as the shortest instructions text has 5 sentences. For identifying relations between events and states we used a lag of 1, as the event and the change of state are usually described in the same sentence or in following sentences. For identifying relations between events without considering the object, we also took a lag of 1, because in texts with longer sentences, the test tends to discover false positives when applied with a longer lag. Furthermore, to reduce the familywise error rate during the multiple comparisons, we decreased the significance threshold by applying the Bonferroni correction.

To compare the approach with that of using grammatical patterns, we implemented patterns with a causal link that contain words such as *until*, *because*, *before*, etc. We also added the conjunction *and* to the causal links, as it was often used in the recipes to describe a sequence of events. We also implemented a verb-noun-adjective pattern to search for the relation between events and states, and a verb(present)-noun-verb(past) pattern to search for relations between events and states. Finally, we implemented a conditional pattern (e.g. the *if-then* construction). As an input for these patterns we used once again the text instructions with POS tags from the Stanford Parser.

5 Results

The human expert discovered an average of 25.25 causal relations per text document. Using the grammatical patterns, an average of 4.15 causal relations per text document were discovered. Using the time series approach, an average of 20.9 causal relations per document were discovered.

The number of causal relations discovered in each text document can be seen in Figure 1. It shows that the number of discovered relations is lower in texts with short sentences.

Furthermore, the recall for each textual instruction is shown in Figure 2. The recall increases with decreasing the sentence length, while the false discovery rate (FDR) decreases.

On the other hand, the recall for the grammatical patterns is low for all instructions. However, in

[8] `http://nlp.stanford.edu/software/lex-parser.shtml`

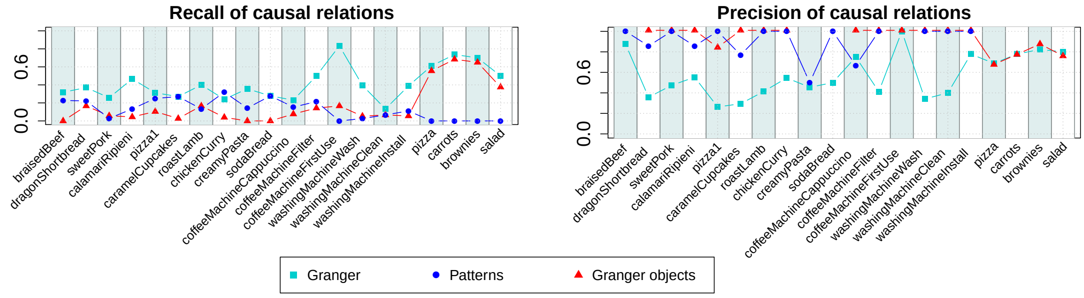

Figure 2: Recall and precision of the discovered causal relations for each dataset. Square indicates Granger causality, circle grammatical patterns, triangle Granger causality when using only the event-object pairs.

difference to the time series approach, the grammatical patterns have a high precision.

The precision and recall of the time series when using only the event-object pairs (Algorithm 1) show that the precision for the event-object pairs is very high in comparison to the overall time series precision (Figure 2).

Finally, we tested whether there is a significant correlation between the performance of the approaches and the type of textual instruction. We applied a two sided correlation test that uses the Pearson's product moment correlation coefficient. The results showed that in the approach using time series and the Granger causality test, the performance is inversely proportional to the sentence length and the number of events in the sentence. On the other hand, the approach using the grammatical patterns is proportional to the sentence length and the number of events.

6 Discussion

In this work we presented an approach that relies on time series to discover causal relations in textual descriptions such as manuals and recipes.

Among the advantages of the approach are the following. The approach allows the discovery of implicit causal relations in texts where explicit causal relations are not discoverable through grammatical patterns. It does not require a training phase (assuming the text has POS tags), or explicit modelling of grammatical patterns. This makes the approach more context independent. It discovers relations different from those discovered with grammatical patterns, and can detect causal relations between elements that are several sentences apart. This indicates that both approaches can be combined to provide better performance.

Apart from the advantages, there are several shortcomings to the approach. The approach is not suitable for texts with complex sentence structure and many events in one sentence, as this generates false positive relations. The cause for this is that when we have several words we want to test in the same sentence, they will also have the same time stamp. To solve this problem, one can introduce additional sentence separators.

Another characteristic of textual instructions is that they often omit the direct object. On the other hand, as the results showed, the usage of objects reduces the generation of false positives. To make use of this, we can introduce a preprocessing phase, where verbs that are in conjunction all receive the same direct object.

Another problem is the lag size in the Granger causality test. The test is very sensitive to the lag size in the case when it is applied to events that do not have direct objects. On the other hand, the approach is less sensitive to the lag when the sentence length is reduced, and it is robust when direct object is used.

Another problem associated with the Granger causality test is whether it discovers causality or simply correlation. As the approach does not rely on contextual information, apart from the causes, it also discovers any number of correlations in the time series. To that end, Granger causality is probably not the best tool for searching for causal relations in textual instructions, but it produces results in situations where the grammatical patterns are not able to yield any results.

As a conclusion, the usage of time series in textual instructions allows the discovery of implicit causal relations that are usually not discoverable when using grammatical patterns. This can potentially improve the learned semantic structure of ontologies representing the knowledge embedded in textual instructions.

References

BBC (2015). BBC Food Recipes. Retrieved: 22.04.2015 from http://www.bbc.co.uk/food/recipes/.

Cimiano, P. and Völker, J. (2005). Text2onto. In Montoyo, A., Muńoz, R., and Métais, E., editors, *Natural Language Processing and Information Systems*, volume 3513 of *Lecture Notes in Computer Science*, pages 227–238. Springer Berlin Heidelberg.

Cole, S., Royal, M., Valtorta, M., Huhns, M., and Bowles, J. (2006). A lightweight tool for automatically extracting causal relationships from text. In *SoutheastCon, 2006. Proceedings of the IEEE*, pages 125–129.

Cooper, G. (1997). A simple constraint-based algorithm for efficiently mining observational databases for causal relationships. *Data Mining and Knowledge Discovery*, 1(2):203–224.

Girju, R. (2003). Automatic detection of causal relations for question answering. In *Proceedings of the ACL 2003 Workshop on Multilingual Summarization and Question Answering - Volume 12*, MultiSumQA '03, pages 76–83, Stroudsburg, PA, USA. Association for Computational Linguistics.

Granger, C. W. J. (1969). Investigating Causal Relations by Econometric Models and Cross-spectral Methods. *Econometrica*, 37(3):424–438.

Granger, C. W. J. (2001). Testing for causality: A personal viewpoint. In Ghysels, E., Swanson, N. R., and Watson, M. W., editors, *Essays in Econometrics*, pages 48–70. Harvard University Press, Cambridge, MA, USA.

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

Kim, H. D., Castellanos, M., Hsu, M., Zhai, C., Rietz, T., and Diermeier, D. (2013). Mining causal topics in text data: iterative topic modeling with time series feedback. In *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*, CIKM '13, pages 885–890, New York, NY, USA. ACM.

Kim, H. D., Zhai, C., Rietz, T. A., Diermeier, D., Hsu, M., Castellanos, M., and Ceja Limon, C. A. (2012). Incatomi: Integrative causal topic miner between textual and non-textual time series data. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 2689–2691, New York, NY, USA. ACM.

Li, X., Mao, W., Zeng, D., and Wang, F.-Y. (2010). Automatic construction of domain theory for attack planning. In *IEEE International Conference on Intelligence and Security Informatics (ISI), 2010*, pages 65–70.

Mani, S. and Cooper, G. F. (2000). Causal discovery from medical textual data. In *Proceedings of the AMIA annual fall symposium 2000, Hanley and Belfus Publishers*, pages 542–546.

Radinsky, K., Davidovich, S., and Markovitch, S. (2011). Learning causality from textual data. In *Proceedings of the IJCAI Workshop on Learning by Reading and its Applications in Intelligent Question-Answering*, pages 363–367, Barcelona, Spain.

Sil, A. and Yates, E. (2011). Extracting strips representations of actions and events. In *Recent Advances in Natural Language Processing*, pages 1–8.

Silverstein, C., Brin, S., Motwani, R., and Ullman, J. (2000). Scalable techniques for mining causal structures. *Data Min. Knowl. Discov.*, 4(2-3):163–192.

Wong, W., Liu, W., and Bennamoun, M. (2012). Ontology learning from text: A look back and into the future. *ACM Comput. Surv.*, 44(4):20:1–20:36.

Zhang, Z., Webster, P., Uren, V., Varga, A., and Ciravegna, F. (2012). Automatically extracting procedural knowledge from instructional texts using natural language processing. In Chair), N. C. C., Choukri, K., Declerck, T., Doğan, M. U., Maegaard, B., Mariani, J., Moreno, A., Odijk, J., and Piperidis, S., editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).