Distribution Agreement

Signature:

_Matthew Hausknecht__________________ ______________
[Student's name typed] Date

Heuristic Based Extraction of Causal Relations from Annotated Causal Cue Phrases

by

Matthew J. Hausknecht

Adviser

Eugene Agichtein, Phillip Wolff, Li Xiong

Department of Mathematics and Computer Science

____
Full Name of Adviser
Adviser

____
Full Name of Committee Member 2, typed
Committee Member

____
Full Name of Committee Member 3, typed
Committee Member

____
Date

Heuristic Based Extraction of Causal Relations from Annotated Causal Cue Phrases

By

Matthew J. Hausknecht

Adviser

Eugene Agichtein, Phillip Wolff, Li Xiong

An abstract of
A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Department of Mathematics and Computer Science

2009

Abstract

Heuristic Based Extraction of Causal Relations from Annotated Causal Cue Phrases

By Matthew J. Hausknecht

This work focuses on the detection and extraction of *Causal Relations* from open domain text starting with annotated *Causal Cue Phrases* (CCPs). It is argued that the problem of causality extraction should be decomposed into two distinct subtasks. First, it is necessary to identify Causal Cue Phrases (CCPs) inside of a body of text. Second, using these CCPs, the cause and effect phrases of each causal relation must be extracted. To prove that CCPs are an essential part of causality extraction, it is experimentally demonstrated that the accuracy of cause and effect phrase extraction dramatically increases when CCP knowledge is utilized. A 31% increase in accuracy of cause and effect phrase extraction of two equivalent CRF machine learning algorithms is found when simple, word-based knowledge of CCPs is taken into account. Furthermore, it is shown that cause and effect phrase extraction can be performed accurately and robustly without the aid of complex machine learning techniques. A simple, heuristic based extraction algorithm, centering around three distinct classes of CCPs, is introduced. This algorithm achieves an accuracy of 87% on the task of extracting cause and effect phrases. While the problem of identifying CCPs in open domain text is not addressed, it is hypothesized that this task is far easier than identifying cause and effect phrases alone because the space of all possible CCPs is far smaller than that of all causal relations. Finally, this work contributes a free, publicly accessible corpus explicitly annotated with both intra-sentential causal relations and corresponding Causal Cue Phrases. It is our hope that this resource may see future use as a standard corpus for the task of causality extraction.

Heuristic Based Extraction of Causal Relations from Annotated Causal Cue Phrases

By

Matthew J. Hausknecht

Adviser

Eugene Agichtein, Phillip Wolff, Li Xiong

A thesis submitted to the Faculty of Emory College of Arts and Sciences
of Emory University in partial fulfillment
of the requirements of the degree of
Bachelor of Sciences with Honors

Department of Mathematics and Computer Science

2009

Table of Contents

Introduction

“We do things in the world by exploiting our knowledge of what causes what” (Hobbs 2005). Causality underlies human ability to analyze events in the past and predict events in the future. Because of this, Causal information can greatly improve state of the art Information Retrieval and Question Answering systems by providing a source of structured data to draw upon when answering questions. Specifically, questions involving reasoning such as *why, what,* and *how* stand to benefit most from the addition of causal knowledge to the Question Answering process.

(a) Why did <Event-X> happen?

(b) What will be the consequences of <Occurrence-Y>?

(c) How did <Characteristic-Z>evolve?

Problem Statement

We hypothesize that inclusion of annotated Causal Cue Phrases will increase both accuracy of extracting the cause and effect phrase of implicit and explicit intra-sentential causal relations from open domain text. To test this hypothesis, we train and evaluate two machine learning algorithms on a corpus annotated with causal relations. The two algorithms are identical in all way except one of them has knowledge of Causal Cue Phrases. An increase in extraction accuracy is expected to be observed in the machine learning algorithm which has access to CCP knowledge.

Additionally, we propose a simple, heuristic based method of extracting cause and effect phrases from annotated CCPs. To demonstrate that complex machine learning algorithms are not necessary to extract causal relations with high accuracy, we show the heuristic based extraction algorithm performs quite well on both the main causality corpus and an additional test corpus.

Related Work

Initial attempts at causality extraction involved matching hand coded patterns and performing inference from domain-specific knowledge bases (Kaplan, 1991). Also domain specific, Khoo et al. (2000) focused on extraction of explicit causal relations for the financial news domain by employing hand-crafted linguistic patterns, achieving a precision of 76%. Similarly, Garcia (1997) utilized manual linguistic indicators to extract causality from sentences of French text.

More recently, Girju (2003) used machine learning techniques to extract intra-sentential causal relations based on features derived from WordNet (Miller, 1995) synsets of noun phrase heads. She achieves a maximum precision of 73.91% but only attempts to extract causality between noun phrases signaled by a verbal Causal Cue Phrase.

Chang and Choi (2006) employed unsupervised learning techniques to extract intra-sentential causal relations. They use lexical patterns to find possible causal sentences and then classify each as causal or non-causal. They report an *F*-score of 77.37% when using both word and concept pair probabilities as well as cue phrase probabilities.

Pitler et al. (2008) is sought to identify and classify discourse relations in the Penn Discourse Tree Bank. Pitler et al. refer to causal relations as *contingency relations* and reports and an overall accuracy of 74.74% classifying four primary types of discourse relations and 93.09% accuracy when classifying explicit relations. However, by only considering discourse connectives, many implicit causal relations were missed, lexical causatives.

Other work has focused on the extraction of inter-sentential causation. Marcu and Echihabi (2002) looked for sentence pairs connected with *Because of* and *Thus*, resulting in 57% accuracy classifying marked, explicit inter-sentential relations. Likewise Pechsiri and Kawtrakul (2007) identified long-distance causal relations spanning multiple EDUs (Elementary Discourse Units) and report 88% accuracy.

Higashinaka and Isozaki (2008) concentrate primarily on implementing a Question Answering system for why-Questions. They addressed causality extraction only as a subtask and reported no measures of extraction accuracy.

This work differs from previous work in several ways. First, we place an emphasis on extraction of all intra-sentential causal relations. Past work has focused on explicit, implicit or inter-sentential relations but has not tried to tackle the much more difficult problem of general extraction for all intra-sentential relation. However, unlike previous work, we start with explicit knowledge of annotated causal cue phrases. However, as the primary goal is to prove that CCPs are essential for causality extraction, this remains a valid starting point.

Annotation

Throughout this paper, Causal Relations are annotated in the following manner:

> [Cause Phrase]C [Causal Cue Phrase]R [Effect Phrase]E
>
> [I like this example]E [because]R [it is simple.]C

Each causal relation can be broken down into three fundamental parts: the cause phrase, effect phrase, and causal cue phrase. In our annotation scheme, the cause phrase is surrounded by brackets and followed by a *C*. Likewise, the effect and causal cue phrases are also surrounded by brackets and followed respectively by *E*, and *R*.

Causal Cue Phrases

A Causal Cue Phrase is a word or phrase indicating the presence of a Causal Relation. Cause and effect phrases can range over the nearly limitless set of conceivable states and events. However, structural analysis of English text reveals a limited set of keywords or phrases signaling the presence of a causal relation. For example, in the sentence

[Markets usually get noticed]E [because]R [they soar and plunge.]C

because indicates the existence of the Causal Relation and links the Cause Phrase to the Effect Phrase. We broadly define Causal Cue Phrases to include *explicit causatives* (e.g. because, if, as, since, etc.) and *implicit (lexical) causatives* (e.g. kill, melt, increase, dry, etc.). We find that all intra-sentential causal relations contain a Causal Cue Phrase in one of the former categories.

Causal Relations

The random house dictionary defines causality as denoting a necessary relationship between one event (called cause) and another event (called effect) which is the direct consequence of the first. Similarly, a Causal Relation is defined as the textual encoding of a single cause effect pair. The difficulty of automatically extracting Causal Relations can be explained in part by the complex manner in which causation is encoded in English text. The following distinctions help to understand different categories of Causal Relations:

Scope: Causal Relations can occur inside of a single sentence, between sentences, or between paragraphs of text. Intra-sentential causal relations concern causal relations within a sentence, as illustrated by the following sentence:

[Earthquakes]C[generate]R[tidal waves.]E

Inter-sentential causal relations concern causal relations between sentences, as illustrated by the following sentence:

[The stock market dropped.]C[As a result,]R[investors lost money.]E

While other work focuses on the extraction of long distance causal relations (Pechsiri and Kawtrakul 2007), we are concerned only with intra-sentential causality.

Explicit or Implicit: An *Explicit Causal Relation* contains a distinct, non-overlapping, cause, effect, and Causal Cue Phrase. Examples of explicit causal relations are as follows:

- [Mr. Dinkins attracted many whites]E precisely [because of]R [his reputation for having a cool head.]C
- [The increase in carbon dioxide]E is largely [caused by]R [the burning of fossil fuels.]C
- [The crush]C [led to]R [Manic Monday's worst decline.]E

Implicit Causal Relations, also known as *Lexical Causatives*, contain a separate cause and effect but an overlapping Causal Cue Phrase. Unlike Explicit Causal Relations, this Causal Cue Phrase encodes information necessary to understand the effect phrase. For example, in the sentence,

[The problems in Arizona]C have only [[increased]R our resolve to pass the bill.]E the Causal Cue Phrase *increased* indicates that the cause resulted in *an increase of* resolve to pass the bill. Other examples of Implicit Causal Relations and the rephrasing of their Causal Cue Phrases include:

- [A late market rally]C [[erased]R a 28-pence fall.]
    - E(erased → caused the erasure of)
- [The confusion]C effectively [[halted]R one form of program trading.]E
    - (halted → caused to halt)

We seek to identify and extract all types of intra-sentential causal relations, including both explicit and implicit causal relations. For further discussion on implicit and explicit causal relations, see Wolff et al. (2005).

Ambiguity: Blanco et al. (2008) notes that many Causal Cue Phrases can signal causation in some cases but not in others. The presence of a nearly unambiguous CCP (e.g. *because, caused)* will almost certainly signal the existence of a causal relation. However, more ambiguous CCPs (e.g. *since, as, and*) may or may not indicate causality. We find that all CCPs possess varying degrees of ambiguity and a steady gradient exists between the most and least ambiguous.

Blanco et al. (2008) also distinguishes *Marked* or *Unmarked* causal relations; marked causal relations containing a specific linguistic unit or cue phrase and unmarked lacking any cue phrase. We find all intra-sentential causality to be marked and hypothesize the existence of unmarked causal relations exclusively at the inter-sentential level.

Corpus Creation

To the best of the author's knowledge, there is yet no standard corpus specifically for the task of causality extraction. Several past works have manually created corpuses of varying size but none is publicly available. Chang and Choi (2005) bootstrapped a large, raw corpus from 5 million TREC articles. Girju (2003) manually annotated a corpus containing 2000 sentences. Following the same trend, Pechsiri and Kawtrakul (2007) manually annotated a Thai corpus contained 8000 EDU (elementary discourse units) from the agricultural and general health domain. In the same vein, Inui manually annotated a corpus of 750 Japanese news articles. Marcu and Echihabi (2002) used two very large corpuses but only looked for causality between sentences expressed using a small set of keywords.

Other corpuses such as TimeBank (Pustejovsky et al., 2003) contain limited causal annotations but lack annotation for implicit causal sentences. Similarly the Penn Discourse Tree Bank (PDTB) (Miltsakaki et al., 2004) contains contingency relations but lacks much implicit and explicit causality.

Our corpus consists of 2000 unique sentences taken from the Propbank corpus (Palmer et al., 2005). Each causal relation in our corpus contains explicit annotation for the Cause Phrase, Effect Phrase, and Causal Cue Phrase.

Construction Method

Two-thousand unique sentences were randomly taken from the Propbank corpus. All Propbank annotation was removed, leaving only the pure sentence text. Next, these sentences were submitted to Amazon Mechanical Turk in the form of a Turk task. Turks were asked to provide a token by token labeling of the Cause and Effect Phrase in each sentence and to classify each sentence as causal or non-causal. A redundancy of 5 was used, meaning that for each sentence, 5 different Turk annotations were collected.

To generate the annotated corpus, a series of steps was followed: First, a simple majority vote of Turks was used to determine if each sentence was causal or not. While labeling each token by a majority vote would have been possible, there were several difficulties encountered. First, there were three different token classes (cause, effect, and non-causal) so a majority vote was not guaranteed. Second, it was observed that some Turk data was less than ideal – it seemed that some Turks had selected random tokens and other had marked a sentence as causal but had not selected any cause or effect phrase. To compensate for these issues, a user confidence score for each Turks was first computed. This confidence score was based on the percentage of sentences in which the individual Turk had agreed with the majority of Turks as to whether the sentence in question was causal, weighted by also by the total number of sentence he or she had labeled. Using these confidence scores, it was then possible to perform a weighted voting process in which the votes of Turks with higher confidence scores carried greater weight than those with lower confidence.

After the Turk annotation was complete, another pass was made by a manual annotator to verify the accuracy of the Turk annotations. The manual annotator corrected the boundaries of the cause and effect phrases, and in some cases, either added or deleted causal relations. Additionally, the annotator manually tagged every causal relation with a Causal Cue Phrase.

One notable difference in our corpus with respect to previous work is the much higher percentage of causal relations. We find 719 causal relations in 2000 sentences for a 35.95% chance of a sentence containing a causal relation. Past work using a corpus of 2000 sentences only identified 115 as causal (Girju, 2003) for only a 5.75% chance of a sentence containing a causal relation. This difference is mostly likely a result of our attempt to identify all intra-sentential causality rather than just explicit or explicit causal relations. However, we postulate that high percentage of causal relations found in our corpus is representative of the true percentage of intra-sentential causality. Because the corpus was annotated by Amazon Turks, the high number of causal relations reflects a consensus among multiple English-fluent annotators.

Method

We introduce three classes of Causal Cue Phrases, as well as techniques to identify and extract them. These three classes exhaustively account for all intra-sentential CCPs. They were conceptualized by examining trends in where the cause and effect phrase were located with respect to the CCP in many sentences. As an overview, the Eager CCP generally corresponds to a contingency relation with the following structure: [CCP]R [Contingency Event]C , [Consequent Event]E. The Verbal CCP corresponds to implicit or lexical causatives in which the CCP must be included in the effect phrase: [Cause Phrase]C [[Verbal CCP]R rest of effect phrase]E. Note that the Verbal CCP rests inside of the effect phrase. Lastly, the Non-Verbal class of CCPs involves causation linked by CCPs not acting as verbs. For this class, we find that the effect phrase precedes the cause phrase and the CCP is sandwiched between both: [Effect Phrase]E [CCP]R [Cause Phrase]C.

It's important to note that these three CCP classes are identified based on common patterns of cause effect phrase extraction. While the Verbal and Non-Verbal classes are mutually exclusive, the Eager CCP class is not; each Eager CCP could be interpreted as either a Verbal CCP or Non-Verbal CCP depending on whether the CCP contained a verb. This issue will be further addressed in the Extraction Algorithm section.

Background

All of the identification and extraction techniques for the three CCP classes information contained in syntactic parse trees. We employ the Stanford Parser (Klein and Manning, 2003) to create a parse tree of any given sentence. For example, the sentence, *The slack absorbs the pulling strain generated by an earthquake* is converted into the following parse tree:

```
(ROOT
  (S
    (NP (DT The) (NN slack))
    (VP (VBZ absorbs)
      (NP
        (NP (DT the) (VBG pulling) (NN strain))
        (VP (VBN generated)
          (PP (IN by)
            (NP (DT an) (NN earthquake))))))
    (. .)))

det(slack-2, The-1)
nsubj(absorbs-3, slack-2)
det(strain-6, the-4)
amod(strain-6, pulling-5)
dobj(absorbs-3, strain-6)
partmod(strain-6, generated-7)
det(earthquake-10, an-9)
agent(generated-7, earthquake-10)
```

Figure 1: Sample Stanford Parse Tree with Typed Dependencies

As can be seen, the tree has a nested structure with variable amounts of indentation. All trees start from a single root node (ROOT) and can be traversed via *child* links down to the leaf nodes which are individual words of the sentence. We also note that punctuation such as commas and periods are given their own nodes. To climb from a leaf of the tree to the root of the tree, it is possible to traverse *parent* links. Additionally, it is possible to find the *sibling* of any node by looking for other nodes with the same parent and same level of indentation. Each node has an associated label. Some of the most common labels we see are noun phrase (NP) and verb phrase (VP). The complete list of labels can be found at the Penn Treebank II Style of Annotation (Bies, 1995).

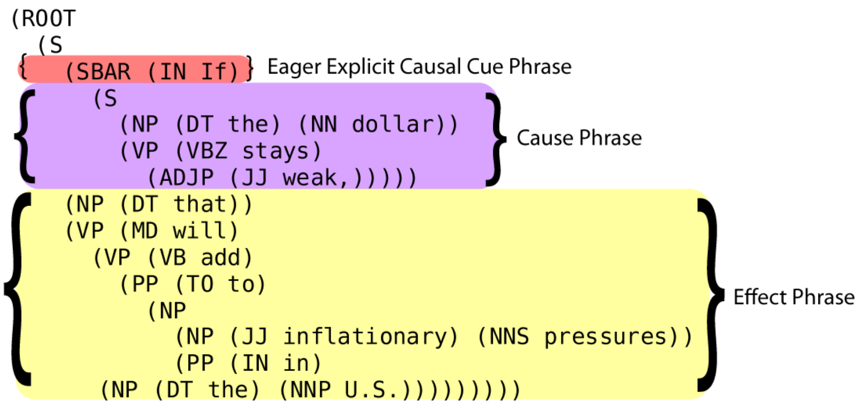
The Stanford Parser also provides information on typed dependencies. These dependencies are listed below the parse tree and represent the relationships between individual words in the parse tree. A complete list of typed dependencies and their meaning can be found in the *Stanford typed dependencies manual* (Marneffe and Manning, 2008). Each typed dependency is structured in the following way:

dependency_name(governing word – index, subordinate word – index)

For example, consider the typed dependency `dobj(absorbs-3, strain-6)`. We can see that the relationship between word 3 in our sentence (absorbs) and word 6 (strain) is that of a direct object. Furthermore, because *absorbs* is the governing word and *strain* the subordinate, we know that *strain* is the direct object of *absorbs*. These typed dependencies can give us much more information about the sentence that the syntactic parse tree alone.

Eager CCPs

[If]R [the dollar stays weak,]C [that will add to inflationary pressures in the U.S.]E



```
(ROOT
  (S
    (SBAR (IN If)
      (S
        (NP (DT the) (NN dollar))
        (VP (VBZ stays)
          (ADJP (JJ weak,)))))
    (NP (DT that))
    (VP (MD will)
      (VP (VB add)
        (PP (TO to)
          (NP
            (NP (JJ inflationary) (NNS pressures))
            (PP (IN in)
     (NP (DT the) (NNP U.S.)))))))))
```

Figure 2: Sentence Matching Eager CCP Class

An *Eager Causal Cue Phrase* is a CCP with an SBAR grand parent, whose children must consist of the following nodes, enumerated in order: comma, a noun phrase (NP), and a verb phrase (VP). Generally, Eager CCPs correspond to contingency relations, expressing an effect whose occurrence is somehow dependent upon a causal event, as follows:

[Causal Cue Phrase]R [Cause Phrase,]C [Effect Phrase.]E

The following observations pertain to the identification of Eager Causal Cue Phrases: Eager Causal Cue Phrases are most commonly located as the first word in a sentence. Additionally because the sentence must contain two distinct phrases, a comma separating the cause and effect phrases must be present. The challenge arises in many sentences which have multiple commas. Additionally, only in a minority of sentences containing a CCP followed by a comma is the CCP classified as Eager. Because of this, more sophisticated identification techniques are required.

To identify an Eager CCP in a sentence, we want to check for three things. First, the at least one of the words in the CCP must have an SBAR parent. The SBAR tag is used to identify a clause introduced by a subordinating conjunction. Second, the SBAR parent must have as a next sibling, a comma, the location of which will be used to identify the boundary between cause and effect phrases. Lastly, the next two siblings of the SBAR parent after the comma must be a NP and a VP. Using this identification pattern we find that Eager CCPs can be identified with high precision.

Cause and effect phrase extraction proceeds as follows: after the location of the comma which splits the sentence into two phrases is found, cause and effect phrase extraction is simple: the Cause Phrase is found to be all words in the sentence between the CCP and the comma divider. The Effect Phrase is found to be all words contained in the NP VP following the comma divider.

Interestingly we do not observe sentences of the form:

[Causal Cue Phrase]R [Effect Phrase,]E [Cause Phrase.]C

This may be a result of the English language forcing a speaker to specify a condition before a consequence.

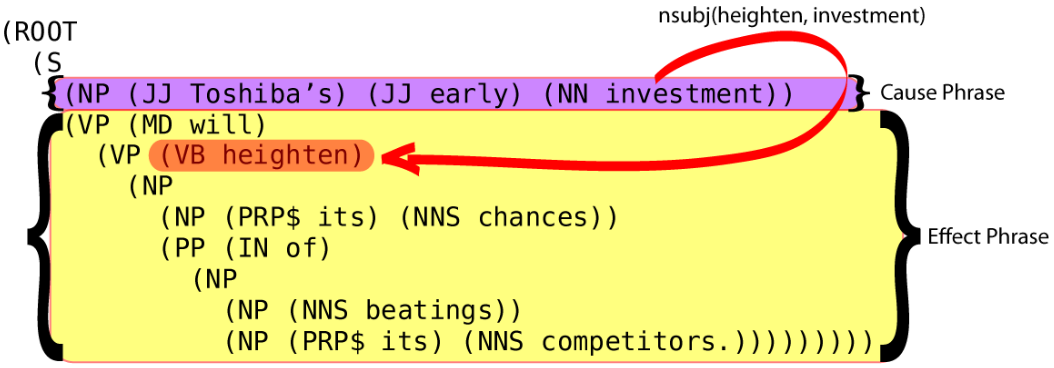
Extraction Pseudocode:

```
// Separates our CP and EP
int comma_location
// Contains the Effect Phrase
Tree NP_Tree, VP_Tree
// The index of the last word in our CCP
int CCP_End_Index

// Check each word in CCP for SBAR grandparent followed by sibling comma, NP, VP
for (each word in CCP)
        Tree grandparent = word.parent().parent()
        if (grandparent == SBAR)
                Tree[] children = grandparent.children()
                if (children[0] == comma && children[1] == NP && children[2] == VP)
                        comma_location = children[0].index()
                        NP_Tree = children[1]
                        VP_Tree = children[2]

// All words from end of CCP to (and including) comma divider
Cause_Phrase = CCP_End_Index ... comma_location
// All words contained in NP_Tree and VP_Tree subtrees
Effect_Phrase = NP_Tree.leaves() + VP_Tree.leaves()
```

Verbal CCPs

An example of a verbal CCP is as follows:

[Toshiba’s early investment]C will [[heighten]R its chances of beating its competitors.]E



```
(ROOT
  (S
 (NP (JJ Toshiba's) (JJ early) (NN investment))
 (VP (MD will)
   (VP (VB heighten)
     (NP
       (NP (PRP$ its) (NNS chances))
       (PP (IN of)
         (NP
           (NP (NNS beatings))
           (NP (PRP$ its) (NNS competitors.)))))))))
```

Figure 3: Sentence Matching Verbal CCP Class

Verbal CCPs are Causal Cue Phrases which are direct parse tree descendents of a Verb Phrase. In practice, Verbal CCPs account for implicit, lexical causatives. Verbal CCPs are identified as CCPs directly descended from a Verb Phrase. Cause and effect phrase extraction is performed in the following manner: the Effect Phrase is identified as the first Verb Phrase parent of the CCP. The Cause Phrase is identified as the largest non-overlapping noun phrase linked to the CCP by an *nsubj* or *nsubjpass* typed dependency. If no such typed dependency exists, the first Noun Phrase preceding the Verb Phrase is used.

Verbal CCPs contain both explicit (e.g. *generated, caused, led to*) and implicit (e.g. *heighten, increase, make*) causal relations. Because we extract the full parent Verb Phrase of the Verbal CCP, the CCP itself is contained in the Effect Phrase. This manner of extraction is ideal for implicit CCPs in which the CCP is needed to understand the Effect Phrase, but is superfluous in explicit CCPs. It is assumed that even in the case of explicit CCPs the inclusion of the CCP in the Effect Phrase does not preclude understanding.

Extraction Pseudocode:

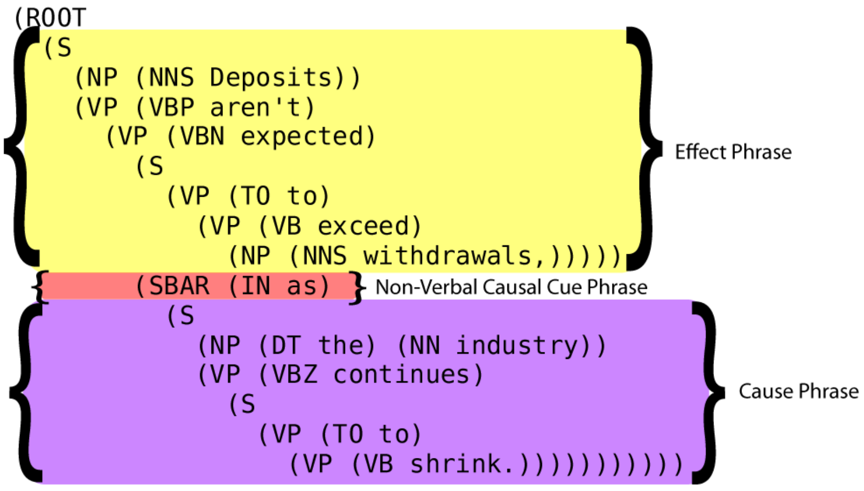
```
// Base of cause and effect phrases
int CP_Head, EP_Head

// Find an nsubj or nsubjpass Typed Dependency which references our CCP
```

```
for (Typed_Dependency TD in Set_of_TypedDependencies)
        if (TD's label == nsubj || nsubjpass)
                // Governer Index of the Typed Dependency
                gov_Index = TD.gov().index()
                // Dependent Index of the Typed Dependency
                dep_Index = TD.dep().index()
                if (CCP contains gov_Index)
                        CP_Head = dep_Index
                        EP_Head = gov_Index

// Find largest NP or S tree not contaning our CCP
Tree cause_tree = CP_Head
while (!cause_tree.overlaps(CCP))
        //expand cause_tree via parent links until root node's label == NP or S
        cause_tree = cause_tree.expandUntil(NP or S)

// Find largest VP tree not overlapping our cause tree
Tree effect_tree = EP_Head
while (!effect_tree.overlaps(cause_tree))
        //expand effect_tree via parent links until root node's label == VP
        effect_tree = effect_tree.expandUntil(VP)

// Find boundaries of each tree
Cause_Phrase = cause_tree.leaves()
Effect_Phrase = effect_tree.leaves()
```

Non-Verbal CCPs

An example of a non-verbal CCP is as follows:

[Deposits aren't expected to exceed withdrawals,]E [as]R [the industry continues to shrink.]C



```
(ROOT
  (S
    (NP (NNS Deposits))
    (VP (VBP aren't)
      (VP (VBN expected)
        (S
          (VP (TO to)
            (VP (VB exceed)
              (NP (NNS withdrawals,))))))
        (SBAR (IN as)
          (S
            (NP (DT the) (NN industry))
            (VP (VBZ continues)
              (S
                (VP (TO to)
                  (VP (VB shrink.))))))))))
```

Figure 4: Sentence Matching Non-Verbal CCP Class

Non-Verbal Causal Cue Phrases are CCPs whose direct parent is not a Verb Phrase. In practice, this includes mostly explicit CCPs such as *because, as,* and *since*. For the purpose of identification, Non-Verbal CCPs are all Causal Cue Phrases which do not have a Verb Phrase parent. Cause and effect phrase extraction is performed in the following manner: to extract the Effect Phrase, we simply look for the first Subject (S) clause preceding our CCP. Cause Phrase extraction involves extracting the first NP or S following but not including the CCP. In the case of the cause phrase, the NP or S found is expanded by traversal of parent links so long as the expanded tree does not include the CCP. The effect phrase is expanded in the same manner but allowed to include the CCP.

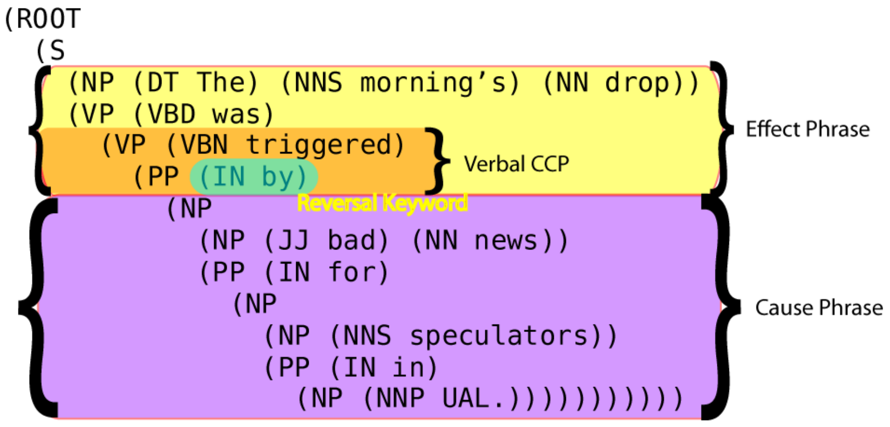
Extraction Pseudocode:

```
// Index of the first word in our CCP
int CCP_Begin

// Find largest NP or S tree not containing our CCP
Tree cause_tree = first_noun_following_CCP
while (!cause_tree.overlaps(CCP))
        cause_tree = cause_tree.expandUntil(NP or S)

// Find largest S tree which is parent of
Tree effect_tree = first_noun_preceding_CCP
while (effect_tree != ROOT)
        effect_tree = effect_tree.expandUntil(S)

// Cause_Phrase is bounded by cause_tree
Cause_Phrase = cause_tree.leaves()

// Effect_Phrase is all words from the start of
// effect_tree up to (not including) the start of our CCP
Effect_Phrase = effect_tree.startIndex() ... CCP_Begin
```

Reversal Keywords

In a minority of Verbal and Non-Verbal CCPs, the cause and effect phrases are switched. This happens primarily in the presence of a small list of keywords. Examples include keywords *by, from,* and *for*. Since the list of reversal keywords is so small, it's easiest to detect when a CCP contains a reversal keyword via a static list and to invert the labeling of cause and effect phrase.

[The morning's drop]E was [triggered by]R [bad news for speculators in UAL.]C



```
(ROOT
  (S
    (NP (DT The) (NNS morning's) (NN drop))
    (VP (VBD was)
      (VP (VBN triggered)
        (PP (IN by)
          (NP
            (NP (JJ bad) (NN news))
            (PP (IN for)
              (NP
                (NP (NNS speculators))
                (PP (IN in)
                  (NP (NNP UAL.))))))))))
```

Figure 5: Example Verbal CCP with Reversal Keyword

The list of verbal reversal keywords is: *for, by, from, attributed,* and *reflecting*

Nonverbal reversal keywords include only the sole keyword *and*.

Extraction Algorithm

The algorithm used to extract cause and effect phrases given causal cue phrases is quite simple:

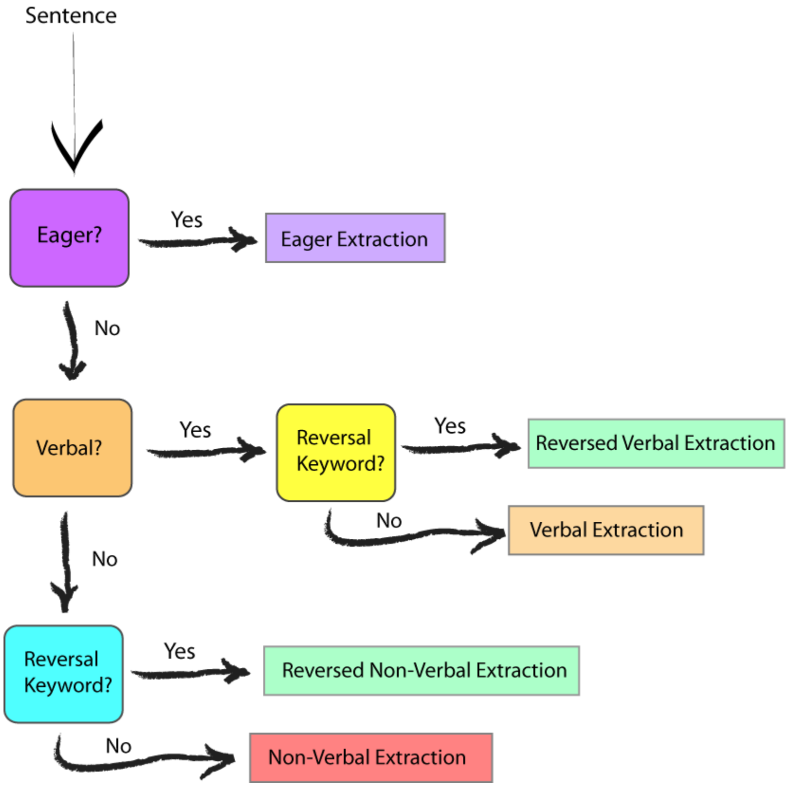


Figure 6: Extraction Algorithm Flow Chart

Because the Eager CCP class is not mutually exclusive with respect to the Verbal and Non-Verbal CCP classes, we put it first on the identification queue. This way, the Eager CCP class has the chance to match itself against all incoming sentences. Because Verbal and Non-Verbal CCP classes are mutually exclusive, the order in which they are allowed to match incoming sentences is unimportant.

Experimentation

To demonstrate the effectiveness of cause and effect phrase extraction when using Causal Cue Phrases, we propose to compare accuracy of two Conditional Random Fields Relations Learning Algorithms (Sutton and McCallum, 2006) trained to perform a token level labeling of cause and effect phrases. One of these CRFs will be trained given access to CCP knowledge in the form of a single additional Boolean feature. In all other respects these two algorithms will be identical.

Because it is only possible to extract correct cause/effect phrases from sentences containing a causal relation, the training set for CRF will consist of all causal sentences in the corpus. Training and testing for CRF will take place using a ten-fold cross validation method. Both algorithms will perform extraction on a token level, classifying each token as one of three distinct classes: cause (part of cause phrase), effect (part of effect phrase), and non-causal (part of neither cause nor effect phrase).

To evaluate the performance of each algorithm, a phrased based accuracy metric will be employed. An algorithm will be said to have correctly identified the cause phrase of a sentence if the cause phrase it identified overlaps with expected cause phrase but does not overlap with the expected effect phrase. Similarly, to correctly identify the effect phrase, it is necessary to overlap with the expected effect phrase but not the expected cause phrase.

Additionally, we employ a token based accuracy metric to provide a stricter evaluation. Each token is said to be correct if matches with the expected class label. One drawback of using such a token based accuracy metric is the ability to get rather high accuracy scores simply by marking every token as Non-Causal. To combat this, we introduce a third accuracy metric, termed *Focused Token Accuracy* which resembles the usual token based accuracy metric only it is computed only over the accepted cause and effect phrases. This further step eliminates all Non-Causal tokens, and reveals how well each algorithm is able to perform attempting to label only cause and effect phrases.

The CRF will be trained with the following feature set:

- Part of Speech: the part of speech of the current token
- Stem: the stem of the current token
- VP or NP: whether the current phrase (most direct parent) is a noun phrase or verb phrase
- Adjacent word information: the words and features of 3 adjacent tokens

For the CRF trained with CCP knowledge, we add a single additional feature to the above set. This feature consists of only two classes; positive indicating that the present word is a member of the CCP and negative, indicating the present word is not a member of the CCP.

In addition to evaluating the performance of the two CRF algorithms, the performance of the heuristic based extraction system will be examined. Like the CRFs, this algorithm will be evaluated on its ability to provide a token based labeling of the cause and effect phrases of each of the 719 causal relations inside the causality corpus used. The evaluation metrics discussed above will also be applied directly to this algorithm's extraction.

Finally, the heuristic based extraction algorithm will also be tested on a smaller, manually annotated corpus independent of the larger causality corpus. This smaller corpus consists of 100 sentences with a total of 56 causal relations. The decision to re-evaluate the heuristic based algorithm was performed in order to relax concerns regarding a possible over-fitting of the larger causality corpus. Similar to previous tasks, the heuristic algorithm is evaluated on phrasal, token and focused token metrics.

Results

Algorithm	Phrasal	Token	Focused Token
CRF (no CCP feature)	137/719 = 19.05%	10067/19628 = 51.29%	5683/11861 = 47.91%
CRF (CCP feature)	361/719 = 50.20%	12650/19628 = 64.49%	7316/11861 = 61.68%
Heuristic Based	629/719 = 87.48%	14010/19628 = 71.38%	9102/11861 = 76.74%
Heuristic Based (independent corpus)	48/56 = 85.71%	1200/1566 = 76.63%	760/968 = 78.51%

Table 1: Cause Effect Phrase Extraction Accuracy by Algorithm

The heuristic based cause and effect phrase extraction method correctly extracted 629 of 719 causal relations for an accuracy of 87.48%. Of the 90 incorrectly extracted causal relations, 9 were of the Eager class, 49 of the Verbal class, and 32 of the Non-verbal class. CRF, trained without knowledge of causal cue phrase, correctly extracted only 137 of 719 causal relations for an accuracy of only 19.05%. The CRF trained with the exact same features as the other plus knowledge of causal cue phrases correctly extracted 361 of 719 causal relations, achieving an accuracy of 50.20%. This is an improvement gain of 31.15% accuracy simply by knowing which tokens serve as annotated CCPs without having any further extraction tools or features. We can see similar but not as pronounced increases in accuracy when examining the token and focused token accuracy metrics. Finally, by examining the performance on the independent data set, we see that the heuristic based algorithm was able to generalize to another corpus while continuing to maintain high accuracy. The specific results show a slight decrease in phrasal accuracy from 87% to 85%, however, both token based accuracy metrics show corresponding increases in accuracy.

Relation Class	Frequency	Phrasal Extraction Accuracy

Eager	96 / 719 = 13%	90.63%
Verbal	381 / 719 = 53%	87.14%
Non-Verbal	242 / 719 = 34%	86.77%

Table 2: Cause Effect Phrase Extraction Accuracy by CCP Class for Heuristic based algorithm

Extraction accuracy for the heuristic based method by CCP class showed relatively high accuracy scores for all three classes of CCP. The frequency scores of the CCP classes reveal that Verbal or implicit causation accounts for just over half of the total causal relations. Non-Verbal CCPs account for approximately one third of the total causation and Eager CCPs, while infrequent, were extracted with quite high accuracy.

These results underscore how useful CCPs are for the task of causality extraction. Past work in the extraction of intra-sentential causal relations has achieved accuracy between 76% and 78% accuracy. While the heuristic based extraction achieves accuracy notably higher than previous attempts, knowledge of annotated CCPs is required, which was not the case in previous work. However, the task faced by our extraction system is significantly harder than in past works as reflected by much higher percentage of causal relations inside of the test corpus. The greatest challenge in comparing to past work is the need for a standard corpus by which different systems can test their accuracy. While this remains absent from the field, it is quite hard to reach a conclusion favoring one system over another.

Additionally, considering that the much of the heuristic based extraction method is founded around information from syntactic parse trees and typed dependencies generated by the Stanford parser (Klein and Manning, 2003), it is hard to exceed the accuracy of the parser itself, approximately 90% depending on the specific PCFG and Dependency model used. Unsurprisingly, most of the remaining errors are accounted for by incorrect parses.

Future Work

While we have demonstrated the extraction of cause and effect phrases given a Causal Cue Phrase, the best method of extracting Causal Cue Phrases from raw text has yet to be addressed. Preliminary results using CRF machine learning algorithm indicate high precision (~75%) but low recall (~30%). While this task is quite challenging, we hypothesize that Causal Cue Phrase extraction is a simpler task than attempting to directly extract cause and effect phrases. This is due to the fact that while cause and effect phrases can range over nearly any combination of noun and verb phrases, causal cue phrases form a much smaller and identifiable set. The most significant obstacle to accurate CCP extraction is the problem of ambiguous CCPs.

Conclusion

The problem of causality extraction from open domain text remains a challenging one. We have proposed the decomposition of this task into the two following subtasks: first, identify Causal Cue Phrases in text. Second, use the identified Causal Cue Phrases to extract the cause and effect phrases of each causal relation.

We hope to have shown that the second subtask can be performed reliably with high accuracy using relatively simple methods. Specifically, we have demonstrated that Causal Cue Phrases can be quite useful for causality extraction from open domain text. Three different class of Causal Cue Phrase, *Eager, Verbal,* and *Non-Verbal* were introduced as well as methods of identification and extraction for each class. Experimentation shows that accuracy of cause and effect phrase extraction rises by over 30% when taking account of CCPs. Additionally, 87.48% accuracy is achieved when using a simple, heuristic based extraction method which was shown to generalize beyond the initial corpus.

A final contribution of this paper is the creation of a free and publicly accessible causality corpus containing 2000 sentences, 719 of which contain causal relations. We hope that this corpus will come to serve as a standard of comparison for different causality extraction systems.

References

Bies, A. (1995). Bracketing Guidelines for Treebank II Style Penn Treebank Project.

Blanco, E., Castell, N., & Moldovan, D. (2008). Causal Relation Extraction. *Language Resources and Evaluation.*

Chang, D.-S., & Choi, K.-S. (2005). Incremental cue phrase learning and bootstrapping method for causality extraction using cue phrase and word pair probabilities. *Information Processing and Management* .

Girju, R. (2003). Automatic Detection of Causal Relations for Question Answering. *41st ACL Workshop on Multilingual Summarization and Question Answering.*

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)* .

Hobbs, J. R. (2005). Toward a Useful Concept of Causality for Lexical Semantics. *Journal of Semantics* .

Khoo, S. (2000). Extracting Causal Knowledge from a Medical Database Using Graphical Patterns. *ACL.* Hong Kong.

Klein, D., & Manning, C. (2002). Fast Exact Inference with a Factored Model for Nautral Language Parsing. *Advances in Neural Information Processing Systems 15* .

Marcu, D., & Echihabi, A. (2002). An Unsupervised Approach to Recognizing Discourse Relations. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL).*

Miltsakaki, E., & Joshi, A. (2004). Annotating Discourse Connectives and their Arguments. *HLT/NAACL Workshop on Frontiers in Corpus Annotation.*

Palmer, M., Gildea, D., & Kingsbury, P. (2005). The Proposition Bank: An Annotated Corpus of Semantic Roles. *Computational Linguistics* .

Pechsiri, C., & Kawtrakul, A. (2007). Mining Causality from Texts for Question Answering. *IEICE Transactions on Information and Systems* .

Pitler, E., Raghupathy, M., Mehta, H., Nenkova, A., Lee, A., & Joshi, A. (2008). Easily Identifiable Discourse Relations. *Conference on Computational Linguistics (coling).*

Pustejovsky, J., Gaizauskas, R., & Katz, G. (2003). TimeML: Robust Specification of Event and Temporal Expressions in Text. *IWCS-5, Fifth International Workshop on Computational Semantics.*

Sutton, C., & McCallum, A. (2006). *An Introduction to Conditional Random Fields for Relational Learning.* MIT Press.

Takashi Inui, M. O. (2005). Investigating the Characteristics of Causal Relations in Japanese Text. *Association for Computational Linguistics (ACL) Workshop on Frontiers in Corpus Annotations .*

Wolff, P., Klettke, B., Ventura, T., & Song, G. (2005). Categories of causation across cultures. In *Categorization inside and outside of the lab: Festschrift in hono of Douglass L. Medin.*