The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Including New Patterns to Improve Event Extraction Systems

Kai Cao,[†] Xiang Li,[†] Weicheng Ma,[‡] Ralph Grishman,[◇]
[†]Cambia Health Solutions
[‡]Boston University
[◇]New York University
[†]{kai.cao, xiang.li}@cambiahealth.com, [‡]wcma@bu.edu, [◇]grishman@cs.nyu.edu

Abstract

Event Extraction (EE) is a challenging Information Extraction task which aims to discover event triggers of specific types along with their arguments. Most recent research on Event Extraction relies on pattern-based or feature-based approaches, trained on annotated corpora, to recognize combinations of event triggers, arguments, and other contextual information. However, as the event instances in the ACE corpus are not evenly distributed, some frequent expressions involving ACE event triggers do not appear in the training data, adversely affecting the performance. In this paper, we demonstrate the effectiveness of systematically importing expert-level patterns from TABARI to boost EE performance. The experimental results demonstrate that our pattern-based system with the expanded patterns can achieve 69.8% (with 1.9% absolute improvement) F-measure over the baseline, an advance over current state-of-the-art systems.

Introduction

Event Extraction involves the extraction of particular types of events along with their arguments. In keeping with the design of the ACE (Automatic Content Extraction) event task, we will associate each event mention with a *trigger*, which is a word or a sequence of words (most often a single verb or nominalization) that expresses that event. More precisely, our task involves identifying event triggers and classifying them into specific types. For instance, according to the ACE 2005 annotation guidelines[1], in the sentence "*She was **killed** in an automobile accident yesterday*", an event extraction system should be able to recognize the word "*killed*" as a trigger for the event DIE. This task is quite challenging, as the same event might appear in the form of various trigger expressions and an expression might represent different events in different contexts.

Most recent research work on the ACE Event Extraction task relies on pattern-based or feature-based approaches, creating classifiers for trigger labeling. Since the distribution of ACE event types in the corpus is skewed, the test data includes some relatively common event triggers that do not occur in the training data. To overcome this problem, we propose to use information from other expert-level patterns to help generate more patterns for boosting EE performance. These patterns will be built from the combination of ACE training data and the patterns from external programs, such as the *TABARI* corpus.

Because different EE systems classify events differently, a fully automatic combination would introduce some noise into the training process. On the other hand, a manual review of the entire vocabulary from the external pattern set would be quite burdensome. We have found an effective compromise which does automatic filtering and partial alignment of event classes, with the remainder of the alignment done by hand. The experimental results demonstrate that our pattern-based system with the new patterns can achieve 69.8% (with 1.9% absolute improvement) F-measure over the baseline, an advance over the state-of-the-art systems.

The paper is organized as follows: we first present the background information and definitions used in the following sections of this paper. The background information introduces the baseline system and the expert-level group of patterns we are going to use: the TABARI patterns. In the sections which follow, we describe the framework of generating new patterns from the combination of ACE event patterns and TABARI event patterns, include the list of stop words[2] used in this paper and the corresponding description of each stop word list, present experimental results as well as detailed discussion. In the last two sections we compare our approach with related work and conclude this work and list our future research directions.

Background & Definitions

This research combines TABARI and ACE event patterns. Therefore we need to start by introducing TABARI event patterns and some basic definitions.

Event Extraction Tasks

TABARI (Textual Analysis By Augmented Replacement Instructions)[3] is an open source pattern-based event extraction system. TABARI event patterns [4] are used to extract

[1]https://www.ldc.upenn.edu/site/www.ldc.upenn.edui/files/english-events-guidelines-v5.4.3.pdf

[2]https://github.com/carson529/EventDetectionWithPatterns/tree/master/stopwords

[3]http://eventdata.parusanalytics.com/software.dir/tabari.html

[4]http://brenocon.com/tabari_cameo_verbs.html

events in the TABARI system. Events in the TABARI system include 20 event types and 324 event subtypes. Basic information on the twenty event types is shown in the table below.

From the event names we can see that the event definitions in TABARI are significantly different from ACE events. Therefore the incorporation of TABARI patterns with ACE patterns to perform ACE event extraction is not straightforward, and requires both automatic processing and manual effort.

Baseline System Description

Training proceeds in three passes over the annotated training corpus. Pass 1 collects all the event patterns, where a pattern consists of a trigger and a set of arguments along with the path from the trigger to each argument; both the dependency path and the linear sequence path (a series of noun chunks and words) are recorded. Pass 2 records the frequency with which each pattern is associated with an event type – the ‘event score’. Pass 3 treats the event score as a feature, combines it with a small number of other features and trains a maximum entropy model.

At test time, to classify a candidate trigger (any word which has appeared at least once as a trigger in the training corpus), the tagger finds the best match between an event pattern and the input sentence and computes an event score. This score, along with other features, serves as input to the maximum entropy model to make the final EE prediction. We can see from Table 3 that the resulting system performance is competitive with other recent system results, such as the joint beam search described in (Li, Ji, and Huang 2013).

Definitions

Definition 1 ***TABARI pattern words*** *are all words included in TABARI patterns. Each word combines with a list of TABARI event subtypes, which means that the word appears in at least one pattern of those TABARI event subtypes. A TABARI event‘s pattern words are all words included in the patterns of this TABARI event type.*

Definition 2 ***Irrelevant words*** *are all the words that are never going to be ACE event triggers.* Irrelevant words *include the words of certain part of speech that cannot be ACE event triggers, like prepositions, and all the stop words.*

Definition 3 ***ACE candidate triggers*** *are the words considered to be ACE event triggers. These words should not be ACE event triggers appearing in the training data. Therefore ACE candidate triggers are “New” event triggers to be added to the ACE training data. Each ACE candidate trigger combines with an ACE event type and event subtype.*

Definition 4 ***ACE candidate patterns*** *are ACE event patterns generated from the original ACE event patterns (extracted from the ACE training data) and the list of* ACE candidate triggers.

Definition 5 ***ACE training patterns*** *are ACE event patterns extracted from the ACE training data.*

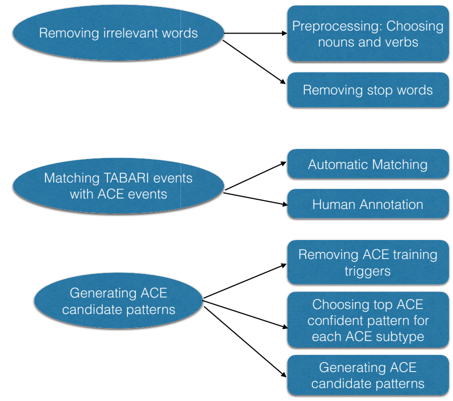

Figure 1: Generating ACE candidate patterns from TABARI event patterns

Definition 6 ***ACE training triggers*** *are ACE event triggers from the ACE training data.*

Definition 7 ***ACE confident patterns*** *are event patterns extracted from the ACE training data with a high evaluation score (In other words, with high confidence) .The* ***ACE confident pattern*** *of an ACE event subtype is the ACE confident pattern with the highest evaluation score.*

Definition 8 ***The similarity of two word sets (or word lists)*** *is the size of intersection of the two word sets.*

Definition 9 ***The similarity between a word list (A) and a group of word lists (G)*** *is the average of similarity between A and each word list in G.* ***The similarity between a TABARI event type A and ACE data*** *is defined as the average of the similarity between A’s pattern words and each ACE event type’s triggers.*

Framework

Our goal for this paper is to improve the performance of original event extraction approach with a group of expert-level patterns from the TABARI program. We call these patterns “TABARI patterns”. This research includes 2 steps:

1. Generating ACE candidate patterns from TABARI event patterns
2. Using ACE candidate patterns to improve the performance of the original event extraction

Generating ACE candidate patterns from TABARI event patterns

The detailed information is shown in Figure 1. This step includes 3 substeps shown in Figure 1 and described below:

Removing irrelevant words from the TABARI pattern words. We call the new smaller list *Reduced TABARI pattern words*. There are 2988 TABARI pattern words. After removing irrelevant words using a set of stop word lists, 224 reduced TABARI pattern words remain. The details will be shown in the next section.

Event types	Event Code	*# of event subtypes*	*# of patterns*
Make public statement	01	10	1075
Appeal	02	25	574
Express intent to cooperate	03	26	1074
Consult	04	7	264
Engage in diplomatic cooperation	05	8	385
Engage in material cooperation	06	5	125
Provide aid	07	6	286
Yield	08	25	391
Investigate	09	5	129
Demand	10	25	179
Disapprove	11	11	543
Reject	12	26	462
Threaten	13	22	505
Protest	14	26	173
Exhibit military posture	15	5	191
Reduce relations	16	13	348
Coerce	17	12	317
Assault	18	13	85
Fight	19	7	431
Attack with weapons of mass destruction	20	7	5

Table 1: TABARI events

Matching TABARI events with ACE events TABARI events include 20 event types and 324 event subtypes. ACE has 8 event types and 33 event subtypes. However they belong to different event styles. Therefore bridging the difference between event definitions becomes the most difficult part of matching these two types of events.

1. **Automatic Matching** We computed the similarity between ACE event subtypes and TABARI event subtypes. For each TABARI event subtype which remains, we chose the most similar ACE event subtype, as the **matched ACE event subtype**. For example, ACE event subtype "Conlict:Attack" is most similar to TABARI event subtype "Threaten with military force, not specified below" (event subtype code : 138). We can see from the definition that an "Attack" is similar to the meaning of "Threaten with military force".

2. **Human annotation**: choosing the corresponding ACE event type for each reduced TABARI pattern word Each TABARI event subtype which remains is matched with an ACE event subtype. Therefore each word in the *Reduced TABARI pattern words* is combined with an ACE event subtype. Combined with ACE event type and subtypes, these Reduced TABARI pattern words become *ACE candidate triggers*. Some of the reduced TABARI pattern words are combined with more than one TABARI event subtype. Although most of the words combined with multiple event subtypes are removed, some remain.

   It won't be too much work to annotate the ACE patterns manually. Therefore the last part of this step is to choose an ACE subtype from a list of ACE subtypes by hand. For example, the ACE candidate trigger "brief" appears in TABARI patterns of four TABARI event subtypes: 040(Consult), 173(Arrest), 020(Make an appeal or request), 042(Make a visit). These four types' matched ACE event subtypes "Contact:Meet", "Conflict: Attack", "Movement:Transport", and "Movement:Transport", respectively. We can see that the matchings between ACE and TABARI event subtypes are basically correct. The word "brief" appears in these four types of event patterns based on different contexts. Therefore we need to choose a "matched" ACE event subtype for the ACE candidate trigger "brief", which is "Contact:Meet".

   Out of the 224 reduced TABARI pattern words, 49 words match only one ACE event subtype. Therefore we only have to annotate the other 175 reduced TABARI pattern words with the proper matching ACE event subtype.

Generating ACE candidate patterns from ACE candidate triggers & ACE training patterns From the two steps above, we have a list of *reduced TABARI pattern words*. Each word in the list combines with an ACE event subtype. After removing ACE training triggers from the list above, we have a list off *ACE candidate triggers*. However, event triggers by themselves won't help improve the EE performance. The pattern-based event extraction system requires event patterns. Therefore we have to generate new ACE event patterns from ACE candidate triggers. These patterns are called *ACE candidate patterns*. The ACE candidate patterns are generated from the combination of ACE candidate triggers and ACE training patterns, as follows:

1. Removing ACE training triggers from the reduced TABARI pattern words. The remaining words are ACE candidate triggers.

2. Choosing a top ACE confident pattern for each ACE subtype, from the ACE training patterns

3. Choosing a corresponding ACE confident pattern for each ACE candidate trigger, and changing the trigger to the ACE candidate trigger

Using ACE candidate Patterns

Finaly, we combined ACE event patterns and ACE candidate patterns to improve the performance of EE. The processing involves two steps:

1. Identifying events with the original ACE training patterns
2. For the words that can be event triggers but are not identified by ACE training patterns, using ACE candidate patterns to identify the new events.

Filtering TABARI Pattern Words

Event triggers are almost entirely verbs and nouns. Accordingly, of the TABARI pattern words, we keep only those tagged as verbs and nouns.

Moreover, some verbs or nouns cannot be event triggers. For example, subject-controlling verbs such as "plan" or 'want", are normally not event triggers. We have identified nine classes of such "stop" words, mostly based on grammatical criteria; these are listed in Table 2 and described below. These are also removed from the list of TABARI pattern words. Therefore the final list of ACE candidate triggers can be quite small and accurate.

We have identified the following classes of stop words:

1. **Argument-nominalizations:** Some nominalizations are more likely to be event arguments than other nouns. These words are called argument-nominalizations. For example, an "attacker" is always an argument of event "Attack", but it cannot be an event trigger.
2. **Special Nouns** Nouns that do not have any subcategorization are unlikely to be event signals. We produce this list by taking all words in COMLEX which do not have any subcategorization and removing all the words in NOMLEX [5] (which are thus assumed to have some arguments).
3. **Verbs with Small Clause Complements** These are similar to subject raising-verbs. They basically connect a subject with a predicate, but also are not really actions.

   For example, in "I like my meat well-done", "like" connects "my meat" with "well-done", meaning that I like situations in which "my meat is well-done". So "like" does not really trigger any action. There are all different kinds of small clauses, but they all connect some noun phrases with some predicate.
4. **Subject Control** Subject-controlling verbs such as "plan" or 'want", are normally not event triggers. The sentence "The army planned to attack the city ." reports an "*Attack*" event, but the trigger is "attack," not "plan".
5. **Subject Raising** In linguistics, raising is the construction where a given predicate/verb takes a dependent that is not its semantic argument, but rather is the semantic argument of an embedded predicate. In other words, an argument that belongs to an embedded predicate is realized syntactically as a dependent of a higher predicate/verb. Subject Raising is the situation where the lower predicate selects the subject. For example,

$$John\ seemed\ to\ leave\ . \qquad (1)$$

   The sentence above might be an "Movement" event but the word "seem" can never become the trigger. Words like "seem" are called subject raising words.
6. **Transparent Nouns** Most of the transparent nouns are quantifiers. For example, in "thousands of people", the word "thousands" is the quantifier of "people". Event triggers are mostly nouns and verbs. Most of the noun event triggers contain almost all the information to identify the events. For example, the noun "war" is probably a "Conflict:Attack" event describing a war. "Appointment" must contain the information of a meeting between two persons, normally celebrities. Therefore transparent nouns cannot be event triggers. We include the transparent nouns in the lists of stop words.
7. **Money** Amounts of currency, such as dollars or euros, are sometimes event arguments, like the amount of a transaction, but they cannot be event triggers.
8. **Time** Time expressions are similar to money words. A time entity can be an event argument but not an event trigger.
9. **ACE Event Arguments** Normally Ace Entities are not event triggers. For example, in the sentence "50 civilians were killed in the attack", the event trigger is the word "kill" while "civilians" is the event argument.

[5] http://nlp.cs.nyu.edu/nomlex/

Experiments

In this section, we will introduce the evaluation dataset, compare the performance of applying pattern expansion with other state-of-the-art systems, and discuss the contribution of pattern expansion.

Data

We used the ACE 2005 corpus as our testbed. For comparison, we used the same test set with 40 newswire articles (672 sentences) as in (Ji and Grishman 2008; Liao and Grishman 2010) for the experiments, and randomly selected 30 other documents (863 sentences) from different genres as the development set. The remaining 529 documents (14,840 sentences) are used for training.

Regarding the correctness criteria, following the previous work (Ji and Grishman 2008; Liao and Grishman 2010; Ji and Grishman 2011; Li, Ji, and Huang 2013), a trigger candidate is counted as correct if its event subtype and offsets match those of a reference trigger. The ACE 2005 corpus has 33 event subtypes that, along with one class "*None*" for the non-trigger tokens, constitutes a 34-class classification problem in this work. Finally we use *Precision* (*P*), *Recall* (*R*), and *F-measure* (*F1*) to evaluate the overall performance.

Performance Comparison

Table 3 presents the overall performance of the systems with gold-standard entity mention and type information. We can see that our system with information from expert-level patterns can improve the performance over our baseline, and also advances the current state-of-the-art systems.

Stop Words	*Size*	*Examples*
Argument Nominalizations	1,383	advisor, betrayer, sailor
Special Nouns	15,660	chicken, clubhouse, goalkeeper
Verbs with Small Clause Complements	219	remember, worry, love
Subject Control	166	plan, want
Subject Raising	14	be, seem, begin
Transparent Nouns	408	cup, million, team
Money	114	dollars, euros, tax, cash
Time	147	afternoon, Monday, January
ACE event arguments	2,408	rebels, missile, immigrants

Table 2: The size and examples of each word list

Methods	*P*	*R*	*F1*
Sentence-level in (Ji and Grishman 2011)	67.6	53.5	59.7
MaxEnt classifier with local features in (Li, Ji, and Huang 2013)	74.5	59.1	65.9
Joint beam search with local features in (Li, Ji, and Huang 2013)	73.7	59.3	65.7
Joint beam search with local and global features in (Li, Ji, and Huang 2013)	73.7	62.3	67.5
Cross-entity in (Ji and Grishman 2011) †	72.9	64.3	68.3
MaxEnt classifier with local features	70.8	61.4	65.7
AceJet baseline	65.5	70.6	67.9
AceJet system with expert-level patterns (TABARI)	65.2	75.2	**69.8**

Table 3: Performance comparison (%) with the state-of-the-art systems. † beyond sentence level.

Related Work

Although there have been quite a few distinct designs for event extraction systems, most are loosely based on using patterns to detect instances of events, where the patterns consist of a predicate, *event trigger*, and constraints on its local syntactic context. The constraints may involve specific lexical items or semantic classes.

Efforts to improve event extraction performance have focused largely on either improving the pattern-matching kernel or adding new reasonable features. Most event extraction frameworks are feature-based systems. Some of the feature-based systems are based on phrase or sentence level extraction. Several recent studies use high-level information to aid local event extraction systems. For example, (Finkel, Grenager, and Manning 2005), (Maslennikov and seng Chua 2007), (Ji and Grishman 2008) and (Patwardhan and Riloff 2007) tried to use discourse, document, or cross-document information to improve information extraction. Other research extends these approaches by introducing cross-event information to enhance the performance of multi-event-type extraction systems. (Liao and Grishman 2010) use information about other types of events to make predictions or resolve ambiguities regarding a given event. (Li, Ji, and Huang 2013) implements a joint model via structured prediction with cross-event features. (Li et al. 2015) improved the event detection performance by exploiting the semantic knowledge encoded in Abstract Meaning Representation.

There have been several efforts over the past decade to develop semi-supervised methods for learning such pattern sets. One thread began with Riloff's observation that patterns occurring with substantially higher frequency in relevant documents than in irrelevant documents are likely to be good extraction patterns (Riloff 1996). (Sudo, Sekine, and Grishman 2003) sorted relevant from irrelevant documents using a topic description and information retrieval engine. (Yangarber et al. 2000) and (Yangarber 2003) developed a bootstrapping approach, starting with some seed patterns, using these patterns to identify some relevant documents, using these documents to identify additional patterns, etc. This approach was further refined in (Surdeanu, Turmo, and Ageno 2006), which explored alternative pattern ranking strategies. An alternative approach was adopted in (Stevenson and Greenwood 2005), which used Wordnet-based similarity to expand an initial set of event patterns. (Huang and Riloff 2012) developed a bootstrapping system to discover new triggers with selected roles. For example, the word "*sniper*" is very likely to be the *agent* of a *Die* event. (Cao et al. 2015) used active learning to fill gaps in the ACE event training data

Syntactic approaches such as Dependency Regularization have also been utilized before to improve the performance of detecting event triggers with specific types. (Cao, Li, and Grishman 2015) used 3 types of dependency regularizations: Verb Chain Regularization, Transparent Regularization, and Nominalization Regularization. (Cao, Li, and Grishman 2016) and (Cao 2016) proposed more Dependency Regularization steps to improve the performance of the Event Extraction framework, including Passive Voice Regularization and Relative Clause Regularization.

Neural networks have been applied to EE recently. (Nguyen and Grishman 2016) proposed to improve the current CNN models for ED by introducing the non-consecutive convolution. developed a hybrid neural network to capture both sequence and chunk information from specific contexts, and use them to train an event detector for multiple languages without any manually encoded features.

Conclusion and Future Work

To date, the use of supervised methods for creating event extractors has been limited by their poor performance even using large annotated training corpora. In this paper, we demonstrate the effectiveness of combining ACE event patterns and TABARI event patterns to generate new patterns to boost Event Detection performance. Since these newly generated patterns may never appear in the training data, they can complement the patterns generated from the original training data to enhance EE performance. The experimental results show that our pattern-based system with the expanded patterns can achieve 69.8% (with 1.9% absolute improvement) F-measure over the baseline, an advance over current state-of-the-art systems.

The TABARI program involves mainly events about political and military activities. However, ACE events also includes commercial information. In the future work, we are going to include event patterns of commercial events from other programs. *GDELT* is an online pattern-based event extraction system. It includes more general information than the TABARI program. The future work will be to improve ACE event extraction performance with the GDELT events. Since GDELT is much larger than TABARI patterns, more information will be extracted to help improve ACE event extraction. Since GDELT is large, more automatic and human work will be added to remove the irrelevant information. This should allow us to improve the efficiency of our active learning by avoiding less promising examples and to improve final EE performance by including triggers not present in the training set.

Deep neural networks have been implemented in the EE task, mostly in well-developed models such as Convultional Neural Networks. Novel approaches emerges in information extraction and other AI tasks. (Tai, Socher, and Manning 2015) introduced the Tree-structured Long Short Term Memory, a generalization of LSTMs to tree-structured network topologies. (Ma et al. 2018a) introduced Seq2Tree networks and (Ma et al. 2017) applied Seq2Tree to a signal processing problem. (Ma et al. 2018b) applies the Seq2Tree network to multimedia signal modeling problems. In the future we expect to investigate the tree-structured LSTMs and the Seq2Tree model for the EE task.

References

Cao, K.; Li, X.; Fan, M.; and Grishman, R. 2015. Improving event detection with active learning. In *Proceedings of RANLP*.

Cao, K.; Li, X.; and Grishman, R. 2015. Improving event detection with dependency regularization. In *Proceedings of RANLP*.

Cao, K.; Li, X.; and Grishman, R. 2016. Leveraging dependency regularization for event extraction. In *Proceedings of FLAIRS*.

Cao, K. 2016. *IMPROVING EVENT EXTRACTION : CASTING A WIDER NET*. Ph.D. Dissertation, New York University.

Finkel, J. R.; Grenager, T.; and Manning, C. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of ACL*.

Huang, R., and Riloff, E. 2012. Bootstrapped training of event extraction classifiers. In *Proceedings of EACL*.

Ji, H., and Grishman, R. 2008. Refining event extraction through cross-document inference. In *Proceedings of ACL*.

Ji, H., and Grishman, R. 2011. Using cross-entity inference to improve event extraction. In *Proceedings of ACL*.

Li, X.; Nguyen, T. H.; Cao, K.; and Grishman, R. 2015. Improving event detection with abstract meaning representation. In *Proceedings of ACL-IJCNLP Workshop on Computing News Storylines (CNewS 2015)*.

Li, Q.; Ji, H.; and Huang, L. 2013. Joint event extraction via structured prediction with global features. In *Proceedings of ACL*.

Liao, S., and Grishman, R. 2010. Using document level cross-event inference to improve event extraction. In *Proceedings of ACL*.

Ma, W.; Cao, K.; Ni, Z.; Ni, X.; and Chin, S. 2017. Sound signal processing based on seq2tree network. In *Proceedings of Interspeech workshop on Vocal Interactivity in-and-between Humans, Animals and Robots*.

Ma, W.; Cao, K.; Ni, Z.; Chin, P.; and Li, X. 2018a. Seq2tree: A tree-structured extension of lstm network. In *Proceedings of LREC*.

Ma, W.; Cao, K.; Ni, Z.; Li, X.; and Chin, P. 2018b. Tree structured multimedia signal modeling. In *Proceedings of FLAIRS*.

Maslennikov, M., and seng Chua, T. 2007. A multi-resolution framework for information extraction from free text. In *Proceedings of ACL*.

Nguyen, T. H., and Grishman, R. 2016. Modeling skip-grams for event detection with convolutional neural networks. In *Proceedings of EMNLP*.

Patwardhan, S., and Riloff, E. 2007. Effective information extraction with semantic affinity patterns and relevant regions. In *Proceedings of EMNLP*.

Riloff, E. 1996. Automatically generating extraction patterns from untagged text. In *Proceedings of AAAI*.

Stevenson, M., and Greenwood, M. 2005. A semantic approach to ie pattern induction. In *Proceedings of ACL*.

Sudo, K.; Sekine, S.; and Grishman, R. 2003. An improved extraction pattern representation model for automatic ie pattern acquisition. In *Proceedings of ACL*.

Surdeanu, M.; Turmo, J.; and Ageno, A. 2006. Counter-training in discovery of semantic pattern. In *Proceedings of EACL 2006 Workshop on Adaptive Text Extraction and Mining (ATEM 2006)*.

Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved semantic representations from tree-structured long short-term memory networks. In *Proceedings of ACL*.

Yangarber, R.; Grishman, R.; Tapanainen, P.; and Huttunen, S. 2000. Automatic acquisition of domain knowledge for information extraction. In *Proceedings of Coling*.

Yangarber, R. 2003. Counter-training in discovery of semantic pattern. In *Proceedings of ACL*.