Metadata of the chapter that will be visualized in SpringerLink

Book Title	Natural Language Processing and Information Systems	
Series Title		
Chapter Title	Interactive Learning with TREE: Teachable Relation and Event Extraction System	
Copyright Year	2015	
Copyright HolderName	Springer International Publishing Switzerland	
Corresponding Author	Family Name	**Tydykov**
	Particle	
	Given Name	**Maya**
	Prefix	
	Suffix	
	Division	
	Organization	Carnegie Mellon University
	Address	Pittsburgh, PA, 15213, USA
	Email	mtydykov@cs.cmu.edu
Author	Family Name	**Zeng**
	Particle	
	Given Name	**Mingzhi**
	Prefix	
	Suffix	
	Division	
	Organization	Carnegie Mellon University
	Address	Pittsburgh, PA, 15213, USA
	Email	mingzhiz@cs.cmu.edu
Author	Family Name	**Gershman**
	Particle	
	Given Name	**Anatole**
	Prefix	
	Suffix	
	Division	
	Organization	Carnegie Mellon University
	Address	Pittsburgh, PA, 15213, USA
	Email	anatoleg@cs.cmu.edu
Author	Family Name	**Frederking**
	Particle	
	Given Name	**Robert**
	Prefix	
	Suffix	
	Division	
	Organization	Carnegie Mellon University
	Address	Pittsburgh, PA, 15213, USA

Email ref@cs.cmu.edu

Abstract	Information extraction, and specifically event and relation extraction from text, is an important problem in the age of big data. Current solutions to these problems require large amounts of training data or extensive feature engineering to find domain-specific events. We introduce a novel Interactive Learning approach that greatly reduces the number of training examples needed and requires no feature engineering. Our method achieves event detection precision in the 80 s and 90 s with only 1 h of human supervision.

Interactive Learning with TREE: Teachable Relation and Event Extraction System

Maya Tydykov(✉), Mingzhi Zeng, Anatole Gershman, and Robert Frederking

Carnegie Mellon University, Pittsburgh, PA 15213, USA
{mtydykov,mingzhiz,anatoleg,ref}@cs.cmu.edu

Abstract. Information extraction, and specifically event and relation extraction from text, is an important problem in the age of big data. Current solutions to these problems require large amounts of training data or extensive feature engineering to find domain-specific events. We introduce a novel Interactive Learning approach that greatly reduces the number of training examples needed and requires no feature engineering. Our method achieves event detection precision in the 80 s and 90 s with only 1 h of human supervision.

1 Introduction

There has recently been considerable progress in the field of event and relation detection and extraction to address the need of acquiring semantic frames from text. This includes the lexical semantic domain (e.g., FrameNet [1]) and the information extraction domain (e.g., MUC [2]). Acquiring frames turns out to be a difficult and unsolved task, and most systems to date have either required manual methods which are expensive and often require expert knowledge [3,4] or are fully automatic but have not been able to achieve high levels of precision [5]. We propose a hybrid system that leverages both automatic techniques and human intervention in order to decrease the amount of human effort needed to introduce a new frame without sacrificing precision.

The goal of our work is to enable an analyst without special linguistic training to teach new events or relations that the system can later extract with high precision. The extracted information is then used to populate a back-end knowledge base. We present a three-stage Interactive Learning approach to teach the system a new event or relation. The pipeline for this process is shown in Fig. 1. In the first step, the teacher introduces the event or relation and the roles she is interested in having extracted. She then annotates a small batch of simple sentences to teach the system this event or relation, using our Assisted Active Teaching module. The next step is active learning, during which the system uses several heuristics to find sentences with a potentially high learning impact (e.g., potentially confusing). The final step is validation, where the system attempts to extract the event in randomly-chosen sentences from a pre-selected corpus and asks the user to correct these. During all three steps, the teacher and the system work together, using the Ontology Builder module, to update a concept ontology which is then used in the extraction process.

© Springer International Publishing Switzerland 2015
C. Biemann et al. (Eds.): NLDB 2015, LNCS 9103, pp. 1–14, 2015.
DOI: 10.1007/978-3-319-19581-0_23

Our main contribution is a simpler, less frustrating way of teaching a system to extract events and relations via an Interactive Learning process. Our process works by progressing from simple to more complex examples, much in the same way that humans are taught new things from childhood to adulthood.

2 Related Work

Common approaches to the problem of event and relation extraction include pattern-matching [6–9], bootstrapping [10], or a combination of the two [11, 12]. A technique common to many of these approaches is to break the event extraction process into two parts, with the first part devoted to detecting event mentions using indicators in the text, and the second part used to extract roles, or arguments [4]. Our system also uses this two-step process for event and relation detection and extraction.

Active learning has previously been shown to be a helpful aid in training information extraction systems. In [13], a user-centric active learning framework called DUALIST was presented, with the goal being to allow for semi-supervised active learning. This framework was applied to text classification tasks with state-of-the-art results. Reference [14] presented an active learning framework for named entity recognition that focused on two techniques - persistent learning and corrective feedback. These two systems had similar goals of reducing user effort while achieving and maintaining high accuracy, and they demonstrate the high potential of applying active learning to various natural language processing tasks. In [15], active learning was applied to train a system to extract noun phrases belonging to specific semantic classes. In [16], it was used to train an event detection system, but focused on finding more positive examples for the user to label rather than negative or potentially confusing ones. Reference [17] used measures of uncertainty and representativeness as their active learning criterion, collecting entity pairs from a corpus of sentences and classifying relations between the pairs, with the training data provided by Amazon Mechanical Turk. Unlike these other works, our system performs both event detection and role extraction for events and relations - a highly difficult task. Our system focuses on the events and relations that the analyst is interested in. Most importantly, in our system, the teacher is an active participant rather than a passive oracle, with the system assisting the teacher in producing good examples.

3 Extraction

Our extraction procedure uses techniques similar to ones used in other systems such as [4], with the event and relation detection process being the first step, and the role extraction process following after that. First, the system annotates a document using the Stanford NLP toolkit to get part of speech tags, dependency parse trees, and co-reference resolution between entities.

The system then filters each sentence for indicators relevant to each type of event or relation. Indicators, often also called triggers in the literature, are groups of items that must be present in the sentence in order for the event

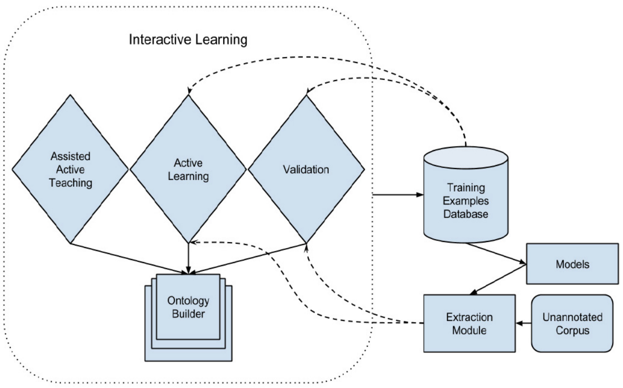


Fig. 1. Current TREE teaching pipeline.

detection process to be initiated, where each item can be the surface form of a word or phrase in the sentence, a named entity type as identified by the named entity recognizer, or a concept from a user-defined set. We build our concept sets interactively using our Concept Builder. Constraints on indicators are event-specific and are added to a teaching database during the teaching process. For example, in an Athlete-Sport relation, a good indicator that the relation may be present in a sentence is if the sentence contains a person and a sport. Thus, in the sentence *"Mary is a hockey player"*, the presence of *"Mary"* and *"hockey"* would satisfy this constraint.

Once an indicator constraint has been satisfied, the system uses a Maximum Entropy classifier and the following features to determine whether or not to trigger an event frame:

1. Features extracted between indicator components:
   (a) Largest word distance
   (b) Largest dependency distance
   (c) Pairwise dependency relations between "types" of indicator components, where types are concepts used to identify the components
2. Total number of indicator components
3. Features extracted with respect to each indicator component:
   (a) Component's number in the sentence paired with POS, NE and text features
   (b) Component's identifying concept paired with POS, NE and text features
   (c) Component's number in the sentence and identifying concept with POS, NE and lexical form features
   (d) POS, NE, and lexical form features are the POS, NE, and text, respectively, of the component, the word to its left, and the word to its right.

If the event classifier labels the sentence as positive, then the sentence becomes a candidate for extraction of event details.

Before proceeding to the role extraction step, the system also consults a negative event trigger classifier. This classifier extracts the same set of features as the first event trigger classifier, but will stop the extraction process if the outcome is positive. For example, in the sentence "John had a heart attack", the teacher can inform the system during the teaching process that the combination of words "heart" and "attack" serve as a negative indicator for an "Attack" event. This step allows for a quick way to eliminate particularly troublesome false positives that may otherwise take a while for the system to learn, minimizing the teacher's time spent and frustration.

In the role extraction step, potential role fillers are filtered in the same way as indicators - by either matching the surface form of a word, a named entity type, or a concept from the relevant ontology sets. Each potential role may have one role indicator. The role indicator can be an item specially selected by the teacher during teaching if the user believes that there is some specific word or concept that can help identify the role. For example, in the sentence *"Karen won a gold medal"*, the word *"medal"* is a good indicator for the role *"Placement"*, which should be filled by the word *"gold"*. To the best of our knowledge, the use of role indicators is novel in our work.

If no such specific role indicator exists or if it is not found in the sentence, the system uses the closest component in the event indicator by distance in the dependency parse tree. If the role and event indicator items are not connected in the dependency parse tree, the system chooses the closest event indicator component by word distance. We refer to the chosen component as the optimal indicator component.

Once the role's optimal indicator component has been identified, the system extracts the following set of features with respect to the pair of role and optimal indicator component:

1. If any indicator component is the same entity as the role filler
2. If the optimal indicator component is a role indicator
3. Features specific to role indicators:
   (a) Dependency relationships between role indicator and the types of indicator components in the sentence; "type" is the concept used to identify the indicator component
   (b) Dependency distance to each type of indicator component
4. Features between optimal indicator component and role filler:
   (a) Dependency relationship
   (b) Dependency distance
   (c) Word distance
   (d) Number of organizations, people, dates, and locations between them
5. If the optimal indicator component is before or after the role filler
6. Dependency relationship between the optimal indicator component and the closest alternative role filler
7. Features extracted for both the optimal indicator component and role filler:
   (a) POS and lexical features, where POS and lexical features are the POS and lexical, respectively, of the item, the word to its left, and the word to its right
   (b) Type of concept used to identify the item.

The system then uses a Maximum Entropy classifier to determine whether or not the pair of optimal indicator component and potential role filler is a good one. Once a set of roles has been identified for the frame, the system makes final role assignments by using the confidence of the role classifier in the case of multiple entities being assigned the same role or multiple roles being assigned to the same entity.

We are currently working under the simplifying assumption of one event frame per event type per sentence, so if a sentence contains multiple potential event indicators for the same event, it needs to choose one of those (along with the roles extracted with respect to that event indicator). Thus, in the final step of the extraction process, the system groups together each potential event indicator in the sentence with the final set of role fillers and uses a Maximum Entropy classifier to get a final classification for the entire event frame. It uses classifier confidences from previous steps as features, specifically:

1. Role extractor classifier confidence.
2. Event detection classifier confidence.

The system will then select the event frame that received the highest final confidence score from the classifier. In the event of a tie, the system uses several rules to rank candidate event frames and chooses the frame with the highest rank.

After performing extraction, the system populates the knowledge base with the events and relations it detects along with the extracted roles. It performs simple string matching to merge the entities by name in the knowledge base.

4 Interactive Learning Process

4.1 Assisted Active Teaching

The teacher starts the process by introducing an event or relation to the system along with the roles she is interested in having extracted. The teacher annotates 10 simple sentences (i.e., short as compared to longer, more convoluted sentences in news articles) that she can either find in relevant documents or come up with herself. During the annotation process, the teacher marks both role fillers and indicators in the sentence. We hypothesize that the teacher can mark indicators as well as role fillers adequately using just common sense (i.e., without special linguistic training). When marking either an indicator or a role filler, the teacher is presented with three kinds of options in order to provide the system with a rule about how to find this indicator or role filler during extraction. These options are the lexical form of the word, the named entity type of the word as recognized by the Stanford NER pipeline with some simple filtering in place to rule out relative dates, or a concept. The teacher selects one of these options, and a new rule to find an indicator or role filler is then added to the teaching database.

4.2 Concept Builder: Adding Concepts to the Ontology

When the teacher selects a concept as the rule to use in identifying a role filler or an indicator, the system works with the teacher to interactively define

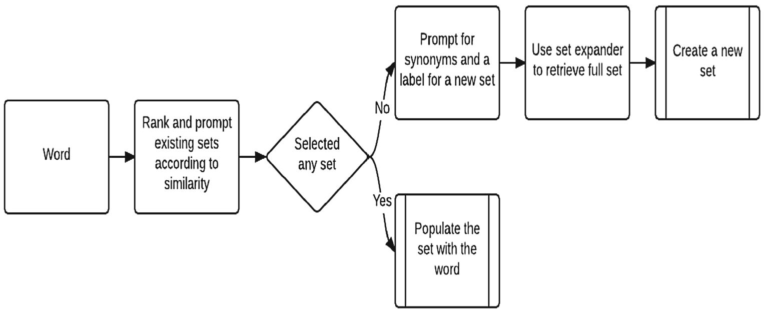

Word
Rank and prompt existing sets according to similarity
Selected any set
No
Yes
Prompt for synonyms and a label for a new set
Use set expander to retrieve full set
Create a new set
Populate the set with the word

Fig. 2. Current Ontology pipeline.

concepts that best fit the user's intention, while simultaneously making each user-added example worth several examples. An important component of our Concept Builder is SEAL, a set expansion tool that automatically scours the web for lists of items and ranks these items to form similar sets [18]. We also created test sets to tune SEAL to reach better accuracy on our corpus. The current work flow of the ontology part of the system is shown in Fig. 2.

When the user wishes to add an indicator or a role label during any of the teaching phases, she is prompted to give the system more information about what makes the particular selection important. One of the choices is a category type. When the user selects the category option, the system can either create a new set or merge the selected entity with an existing set. A new set is created only when the entity is not in any existing sets and the user chooses not to add it to any existing set.

The system first tries to rank existing sets according to the WuPalmer [19] similarity measure based on the depths of the two synsets and the depth of their LCS in the WordNet taxonomies. Then we prompt the user with the ranked sets. If the user chooses any set to merge with, then we further expand the set. The merging process will be discussed later. If the user chooses none of the given sets, then the system prompts the user for two more, similar seed entities to add to the original selected entity. The resulting three seed entities are then sent to SEAL, which expands the category and returns a list of potential additions to the new category. The top K items with the highest belief in the list are then shown to the user, who selects which of these should actually be added to the new set.

The merging process between a user-selected entity and a set in ontology is as follows. When an entity is added to an existing set, the system attempts to further expand the existing set based on the addition of the new entity. It uses several iterations to choose a subset of entities that are already in the set, along with the new entity, as seed entities for further SEAL expansion and then adds the top K entities from the final list of candidates based on a thresholding of SEAL's belief and frequency values for those candidates. The final list is then shown to the user so that she can select which entities will be added to the category being expanded. The merge work flow is shown in Fig. 3.

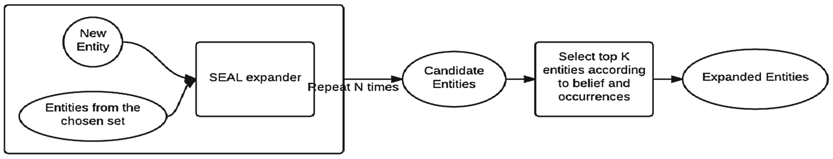

New Entity
Entities from the chosen set
SEAL expander
Repeat N times
Candidate Entities
Select top K entities according to belief and occurrences
Expanded Entities

Fig. 3. Ontology Merge pipeline.

Since the ontologies are built on the fly (as the user teaches the system), there can be points during which there are many distinct sets which are actually very similar to one another. The current implementation uses hierarchical clustering to re-organize the sets in the ontology. The dissimilarity of two entities is defined as the inverse WuPalmer similarity between them. The distance of two sets is defined as the average linkage between the two sets. Initially we set all the existing sets to be in one cluster and then perform hierarchical clustering on these sets. In each step we merge the two clusters with the shortest average linkage and stop when the shortest average linkage between any two sets are above a threshold.

4.3 Active Learning

Once the system has at least 10 examples of the event or relation added via the Assisted Active Teaching process, it can perform active learning for the new event or relation. The goal of the active learning stage is to help the teacher by finding more examples that will be particularly helpful for the system to learn. Finding potential negative examples is one way to achieve this goal. We use several active learning heuristics, several of which are novel, and several of which are commonly used in active learning tasks.

1. Novel heuristics are:
   (a) The system looks through its database of old examples that were taught for other events or relations and tries to extract the new event or relation from those. If it succeeds, it presents up to 5 such sentences to the user for correction. The reasoning behind this heuristic is that while possible, it is unlikely for old teaching sentences containing other events or relations to also contain this new event. Thus, these are good candidates for potential false positives.
   (b) The system looks through a corpus of documents for previously unseen sentences where it is able to extract the event, targeting likely confusing sentences. Conditions to be satisfied are:
      i. There are multiple potential role fillers for a given role.
      ii. There are multiple potential roles for one entity.
2. Standard, confidence-based heuristics are:
   (a) When looking through previously unseen sentences:
      i. Event detection classifier's confidence was less than or equal to .6.

ii. Role extraction classifier's confidence was less than or equal to .6.
iii. Event frame classifier's confidence was less than or equal to .6.

The system looks for up to 10 sentences satisfying any one of the above heuristics and presents them to the user. The user is first shown the indicator used to detect the event and can mark whether or not it is correct. If correct, the system proceeds to ask the user about each role it extracted. The user can mark each role as correct or incorrect. The user can then add any missing roles for the event frame. If the user marks an indicator as incorrect, this adds a negative example for the event detection classifier, the role extraction classifier and the event frame classifier. If the user marks the indicators as correct but some of the roles as incorrect, this adds negative examples for the role extraction classifier. Otherwise, positive examples are added for the classifiers. Finally, the user can add any frames that the system missed in the sentence. The confidence threshold of 0.6 was manually set based on pilot tests, but other confidence thresholds or methods may also be appropriate.

4.4 Validation

The final mode in the 3-step process is validation. In this mode, the system randomly selects 10 sentences from a provided corpus, performs extraction on these sentences, and presents the results to the user for correction. Once the system has finished selecting sentences, the user can correct the system in the same way as described for active learning. The goal of this mode is to pick representative sentences from the corpus for validating the quality of the model trained so far.

5 Experiments

We taught the system 6 events and relations based on a corpus of approximately 75,000 news articles about the 2014 Winter Olympics[1]. A definition of the frames is shown in Table 1. AQ1

We went through 3 cycles of the 3-step process for each event and relation. One cycle consisted of, for each event/relation, teaching the system 10 basic, previously prepared sentences, performing active learning and, finally, performing validation. We used pre-filtered corpora for active learning and validation so that the event or relation in question was more likely to be found. Although one could easily change the order of teaching modes, resulting in a different teaching configuration, we used this cycle because it allowed each event to make use of a relatively large number of sentences taught previously for the other events.

For our baseline, we trained Conditional Random Field (CRF) models using the MALLET toolkit to detect each role for each event and relation, where the roles were identical to the ones defined above. A CRF is a statistical modeling

[1] The corpus is a collection of articles from mainstream English-language press provided by a news aggregator who wished to remain anonymous.

Table 1. XXXX
Definitions of the event and relation frames taught to TREE.

Event/Relation name	Roles
Athlete-country	Athlete, Country
Athlete-sport	Athlete, Sport
Defeat	Winner, Loser, Date, Location, Sport
Withdrawing from competition	Person, Location, Date, Sport
Placing in competition	Person, Location, Date, Sport, Placement
Injury	Person, Location, Date, Body part injured

Table 2. ER = Event Recall; EP = Event Precision; F1 = F1 Score

Event-relation name	ER TREE	ER CRF	EP TREE	EP CRF	F1 TREE	F1 CRF
Athlete-country	0.54	1.0	**0.90**	0.66	0.68	0.79
Athlete-sport	0.77	0.98	**0.90**	0.64	0.83	0.78
Defeat	0.44	0.49	**0.91**	0.71	0.60	0.58
Withdrawing from competition	0.52	0.80	**1.0**	0.71	0.68	0.75
Placing in competition	0.78	0.98	**0.97**	0.68	0.86	0.80
Injury	0.64	0.88	**0.94**	0.72	0.76	0.79

method that takes context into account when making predictions. CRFs are one of the methods commonly used for information extraction [20–22]. We trained the CRF classifier on the same training data produced by the user's interaction with the system and used the following features for training the CRF model:

1. Word lemma
2. POS of the word
3. Named entity type of the word.

For each event and relation, we annotated 50 positive and 50 negative sentences that were not used in training the tested event. For each of these test sets, we then ran the extraction process on each sentence individually, without performing co-reference. TREE was only scored on its results with respect to the event being tested in each test set. Preliminary results for event detection are shown in Tables 2 and 3, with highest precision scores for each test shown in bold.

Scores were calculated only for events and roles of interest for each test set. Event recall was defined as: $R_e = \frac{S_c}{S_a}$, where S_c is the total number of sentences where the system correctly extracted the event, regardless of the correctness of role assignments for the event, and S_a is the total number of sentences where the event was annotated in the test set. Event precision was defined as: $P_e = \frac{S_c}{S_t}$, where S_t was the total number of sentences where the system had extracted the event. Role recall was defined as: $R_r = \frac{R_c}{R_a}$, where R_c was the total number of role fillers that the system got correct in the test set, and R_a was the total

Table 3. RR = Role Recall, RP = Role Precision, F1 = F1 Score

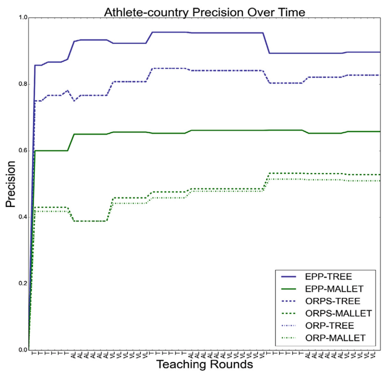
Event-relation name	RR TREE	RR CRF	RP TREE	RP CRF	F1 TREE	F1 CRF
Athlete-country	0.42	0.70	**0.83**	0.51	0.56	0.59
Athlete-sport	0.45	0.60	**0.70**	0.52	0.55	0.56
Defeat	0.21	0.14	**0.52**	0.51	0.30	0.22
Withdrawing from competition	0.42	0.57	**0.82**	0.55	0.55	0.56
Placing in competition	0.47	0.60	**0.72**	0.61	0.57	0.61
Injury	0.48	0.48	**0.79**	0.56	0.59	0.52

number of role fillers annotated in the test set, where a role filler is defined as a span of text paired with a role for the event type. Role precision was defined as: $P_r = \frac{R_c}{R_t}$, where R_t was the total number of role fillers that the system extracted in the sentence for the event frame. The precision trends of both TREE and the CRF are compared in Fig. 4(a)–(f), plotted over the course of each teaching round. The trendlines also include a soft role precision metric which is defined in the same way as the role precision metric except that any overlap in the text between what the system extracted and the gold standard is considered valid.

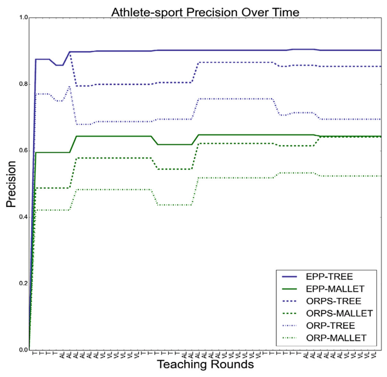
The results from our experiments show that the TREE system can achieve high precision for both event and relation detection and role extraction for most events and relations. Role precision for the Defeat relation is lower than the others, most likely because this is the only relation that, as defined, requires two entities of the same kind to fill two distinct roles (Winner and Loser), which presents both a challenge for the system and an opportunity to explore different teaching methods and configurations. Our system outperforms the CRF classifier in all precision metrics when both are trained on the same, small number of examples. The CRF beats TREE in some recall metrics, but our goal is to maintain high precision. It is also important to note that the CRF had the benefit of training on sentences specifically selected by our Interactive Learning process rather than from randomly selected sentences in a large training corpus. While it is possible that the CRF would outperform our system in precision if trained on a significantly larger dataset, our goal is precisely to avoid the use of such a large amount of data, aiming instead at extracting quality information based on a minimal amount of data.

We were able to achieve these results having spent approximately 5 h teaching the system, which presents a significant advantage over the usual requirement of many person-hours needed to label thousands of examples. Figure 5a shows how the amount of time spent varies with each teaching round (averaged for all events) and Fig. 5b how much time was spent in total on each round for all 6 events and relations.

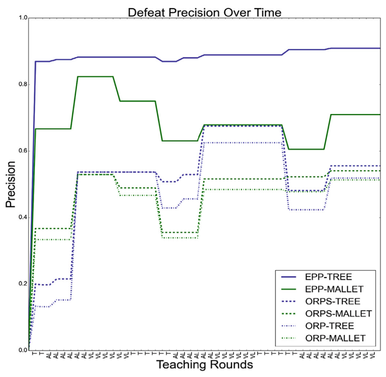
Initially, TREE spends much of its time in the Ontology Builder module, learning how to expand concepts the teacher teaches it. Once it has acquired a sufficient knowledge of these concepts, teaching time decreases. Time spent



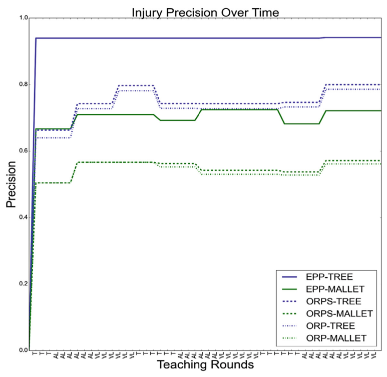
(a)



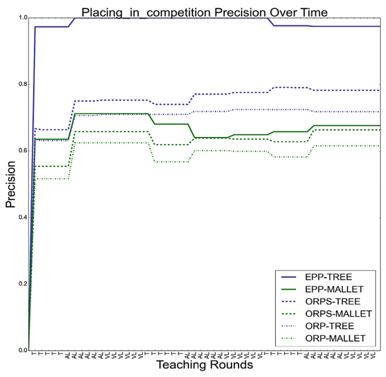
(b)



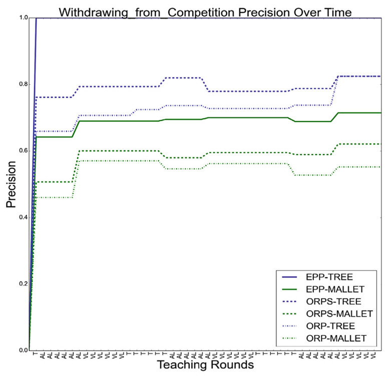
(c)



(d)



(e)



(f)

Fig. 4. Plots of precision over time. Time is measured via teaching rounds. T = Assisted Active Teaching; AL = Active Learning; VL = Validation; EPP = Event Presence Precision; ORPS = Overall Role Precision Soft; ORP = Overall Role Precision.

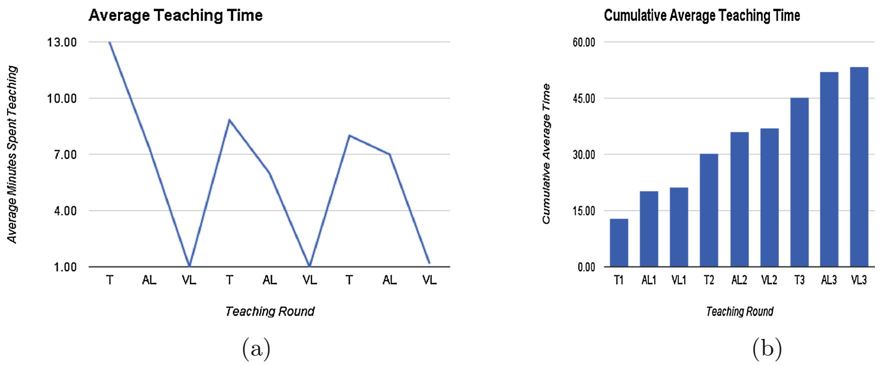


Fig. 5. Graphs of teaching time spent per round.

increases (as in the third round of active learning) if TREE comes across many previously unseen words and uses the Ontology Builder module to expand its concept sets.

6 Conclusion

We believe that the problems of information extraction and, specifically, frame acquisition - particularly when the user wants perform quick data exploration on several events or relations - will not be solvable by fully automated systems or by systems requiring extensive feature engineering. Thus, it is important to explore hybrid methods which can leverage human knowledge while minimizing effort via automated techniques. Our system presents an Interactive Learning technique in which both the system and the user are active participants in the system's learning process. A preliminary evaluation shows that this technique results in reasonable precision. In the future, we wish to explore the optimal ways to configure such hybrid systems as well as what kind of improvement in performance and reduction in effort can be achieved through these systems. Furthermore, we can explore different teaching strategies (i.e., what makes a good or bad teacher). We can also perform more evaluations aimed at testing different parts of the system. For example, we can try to incorporate other, standard corpora used in the Information Extraction domain such as TAC-KBP, MUC, or ACE in order to better compare our work to other work. We can also evaluate the active learning component of our system by comparing it to another system or classifier trained on data via other criteria, e.g. randomly from a relevant corpus. Another direction for future development is to introduce a probabilistic framework into our knowledge base and also into our entity merging procedure, perhaps similarly to the confidence estimation methods described in [23].

Acknowledgments. We would like to thank Jacob Joseph and Eduard Hovy for their valuable advice.

References

1. Baker, C.F., Fillmore, C.J., Lowe, J.B.: The berkeley framenet project. In: Proceedings of the COLING-ACL, pp. 86–90 (1998)
2. Grishman, R., Sundheim, B.: Message understanding conference-6: a brief history. In: Proceedings of the 16th Conference on Computational Linguistics - vol. 1, COLING 1996, pp. 466–471. Association for Computational Linguistics, Stroudsburg (1996)
3. Soderland, S., Fisher, D., Aseltine, J., Lehnert, W.: Crystal inducing a conceptual dictionary. In: Proceedings of the 14th International Joint Conference on Artificial Intelligence, IJCAI 1995, vol. 2, pp. 1314–1319. Morgan Kaufmann Publishers Inc., San Francisco (1995)
4. Ahn, D.: The stages of event extraction. In: Proceedings of the Workshop on Annotating and Reasoning About Time and Events, ARTE 2006, pp. 1–8. Association for Computational Linguistics, Stroudsburg (2006)
5. Vlachos, A., Buttery, P., Séaghdha, D.Ó., Briscoe, T.: Biomedical event extraction without training data. In: Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, pp. 37–40. Association for Computational Linguistics, Boulder, June 2009
6. Grishman, R., Westbrook, D., Meyers, A.: NYU's English ACE 2005 system description. Technical report, Department of Computer Science, New York University (2005)
7. Liao, S., Grishman, R.: Filtered ranking for bootstrapping in event extraction. In: Proceedings of the 23rd International Conference on Computational Linguistics, COLING 2010, pp. 680–688. Association for Computational Linguistics, Stroudsburg (2010)
8. Aone, C., Ramos-Santacruz, M.: Rees: a large-scale relation and event extraction system. In: Proceedings of the Sixth Conference on Applied Natural Language Processing, pp. 76–83. Association for Computational Linguistics, Seattle, April 2000
9. Dzendzik, D., Serebryakov, S.: Semi-automatic generation of linear event extraction patterns for free texts. In: SYRCoDIS 2013, pp. 5–9 (2013)
10. Creswell, C., Beal, M.J., Chen, J., Cornell, T.L., Nilsson, L., Srihari, R.K.: Automatically extracting nominal mentions of events with a bootstrapped probabilistic classifier. In: Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions, pp. 168–175. Association for Computational Linguistics, Sydney, July 2006
11. Xu, F., Uszkoreit, H., Li, H.: Automatic event and relation detection with seeds of varying complexity. In: Proceedings of the 2006 AAAI Workshop on EventExtractionand Synthesis, pp. 12–17 (2006)
12. Huang, R., Riloff, E.: Bootstrapped training of event extraction classifiers. In: Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2012, pp. 286–295. Association for Computational Linguistics, Stroudsburg, (2012)
13. Settles, B.: Closing the loop: fast, interactive semi-supervised annotation with queries on features and instances. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, EMNLP 2011, pp. 1467–1478. Association for Computational Linguistics, Stroudsburg (2011)
14. Culotta, A., Kristjansson, T., McCallum, A., Viola, P.: Corrective feedback and persistent learning for information extraction. Artif. Intell. **170**(14–15), 1101–1122 (2006)

15. Jones, R., Ghani, R., Mitchell, T., Rilo, E.: Active learning for information extraction with multiple view feature sets. In: ATEM-2003 (2003)
16. Altmeyer, R., Grishman, R.: Active Learning of Event Detection Patterns. New York University, New York (2009)
17. Angeli, G., Tibshirani, J., Wu, J.Y., Manning, C.D.: Combining distant and partial supervision for relation extraction. In: EMNLP (2014)
18. Wang, R.C., Cohen, W.W.: Language-independent set expansion of named entities using the web. In: Proceedings of the 2007 Seventh IEEE International Conference on Data Mining, ICDM 2007, pp. 342–350. IEEE Computer Society, Washington, DC, USA (2007)
19. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: Proceedings of the 32nd Annual Meeting on Association for Computational Linguistics, ACL 1994, pp. 133–138. Association for Computational Linguistics, Stroudsburg (1994)
20. Sarafraz, F., Eales, J., Mohammadi, R., Dickerson, J., Robertson, D., Nenadic, G.: Biomedical event detection using rules, conditional random fields and parse tree distances. In: Proceedings of the BioNLP 2009 Workshop Companion Volume for Shared Task, pp. 115–118. Association for Computational Linguistics, Boulder, June 2009
21. Wang, D.Z., Michelakis, E., Franklin, M.J., Garofalakis, M.N., Hellerstein, J.M.: Probabilistic declarative information extraction. In: Li, F. (ed.) ICDE, pp. 173–176. IEEE, Long Beach, CA (2010)
22. Peng, F., McCallum, A.: Accurate information extraction from research papers using conditional random fields. In: HLT-NAACL 2004, pp. 329–336 (2004)
23. Wick, M., Singh, S., Kobren, A., McCallum, A.: Assessing confidence of knowledge base content with an experimental study in entity resolution. In: Proceedings of the 2013 Workshop on Automated Knowledge Base Construction, AKBC 2013, pp. 13–18. ACM, New York (2013)

Author Queries

Chapter 23

Query Refs.	**Details Required**	**Author's response**
AQ1	Please provide captions for Table. 1.	Done

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	• • • under matter to remain	Ⓥ
Insert in text the matter indicated in the margin	⋏	New matter followed by ⋏ or ⋏ⓧ2
Delete	/ through single character, rule or underline or ├──┤ through all characters to be deleted	ℊ/ or ℊ/ⓧ2
Substitute character or substitute part of one or more word(s)	/ through letter or ├──┤ through characters	new character / or new characters /
Change to italics	— under matter to be changed	⌴
Change to capitals	≡ under matter to be changed	≡
Change to small capitals	= under matter to be changed	=
Change to bold type	~ under matter to be changed	~
Change to bold italic	$\overline{\sim}$ under matter to be changed	$\overline{\sim}$
Change to lower case	Encircle matter to be changed	≢
Change italic to upright type	(As above)	⌿
Change bold to non-bold type	(As above)	∤~
Insert 'superior' character	/ through character or ⋏ where required	Y or ⅄ under character e.g. $\overset{2}{Y}$ or $\overset{2}{⅄}$
Insert 'inferior' character	(As above)	⋏ over character e.g. $\underset{2}{⋏}$
Insert full stop	(As above)	⊙
Insert comma	(As above)	,
Insert single quotation marks	(As above)	$\overset{‘}{Y}$ or $\overset{‘}{⅄}$ and/or $\overset{’}{Y}$ or $\overset{’}{⅄}$
Insert double quotation marks	(As above)	$\overset{“}{Y}$ or $\overset{“}{⅄}$ and/or $\overset{”}{Y}$ or $\overset{”}{⅄}$
Insert hyphen	(As above)	├=┤
Start new paragraph	_⌐	_⌐
No new paragraph	⤶	⤶
Transpose	⊔⊓	⊔⊓
Close up	linking ⁀‿ characters	⁀‿
Insert or substitute space between characters or words	/ through character or ⋏ where required	Y
Reduce space between characters or words	| between characters or words affected	T