In Proceedings of the Sixth Workshop on Very Large Corpora, 1998.

An Empirical Approach to Conceptual Case Frame Acquisition

Ellen Riloff and Mark Schmelzenbach
Department of Computer Science
University of Utah
Salt Lake City, UT 84112
riloff@cs.utah.edu, schmelze@cs.utah.edu

Abstract

Conceptual natural language processing systems usually rely on case frame instantiation to recognize events and role objects in text. But generating a good set of case frames for a domain is time-consuming, tedious, and prone to errors of omission. We have developed a corpus-based algorithm for acquiring conceptual case frames empirically from unannotated text. Our algorithm builds on previous research on corpus-based methods for acquiring extraction patterns and semantic lexicons. Given extraction patterns and a semantic lexicon for a domain, our algorithm learns semantic preferences for each extraction pattern and merges the syntactically compatible patterns to produce multi-slot case frames with selectional restrictions. The case frames generate more cohesive output and produce fewer false hits than the original extraction patterns. Our system requires only preclassified training texts and a few hours of manual review to filter the dictionaries, demonstrating that conceptual case frames can be acquired from unannotated text without special training resources.

1 Motivation

Conceptual natural language processing typically involves case frame instantiation to recognize events and role objects in text. For example, an NLP system designed for a business domain might use case frames to recognize business activities such as mergers, acquisitions, or joint ventures. The case frames would contain slots for thematic roles that are associated with each event. For example, case frames for business activities might contain slots for the agents (e.g., companies or people who merge or acquire others) and the objects (e.g., companies that are acquired or products that are being developed).

Unfortunately, acquiring a good set of case frames for a domain can be a major undertaking. Case frames are often lexically indexed so that each case frame is tailored for a specific set of linguistic expressions and their expectations. For example, one case frame might be activated by the phrase "joint venture" and contain slots to recognize the partner companies and objects of the joint venture (e.g., child company or product). A different case frame might be activated by the word "acquisition" and contain slots to recognize the agent (e.g., the acquiring company or person) and the object of the acquisition.

Devising the right set of role assignments for a case frame can be surprisingly difficult. Determining the necessary thematic roles for an event is relatively straightforward, but anticipating how they will be manifested syntactically can be tricky. For example, consider some of the manually defined case frames that were used to recognize terrorist events in the UMass MUC-4 system (Lehnert et al., 1992a).

```
ATTACK (passive-verb "attacked")
Victim      = subject
Target      = subject
Perpetrator = pp(by)
Instrument  = pp(by)

ACCUSATION (active-verb "blamed")
Accuser     = subject
Perpetrator = direct object
Perpetrator = pp(on)

SABOTAGE (noun "sabotage")
Perpetrator = pp(by)
Instrument  = pp(with)
Location    = pp(on)
Victim      = pp(against), pp(of), pp(on)
Target      = pp(against), pp(of), pp(on)
```

The ATTACK case frame shows a very common situation where multiple conceptual roles map to the same syntactic role. When "attacked" is used as a passive verb, the subject may be either a victim or a physical target, and the object of the preposition "by" may be the agent or instrument. It is easy for a person to miss one of these possibilities when defining the case frame manually. The ACCUSATION case frame shows that the same conceptual role can be filled by multiple syntactic roles. For example, the person accused of a crime may be the direct object of "blamed" (e.g., "The government blamed John Smith for the crime") or may be the object of the preposition "on" (e.g., "The government blamed the crime on John Smith"). The SABOTAGE case frame illustrates that a multitude of prepositional arguments may be necessary for some case frames. Prepositional arguments are especially difficult for a person to anticipate when defining case frames by hand.

It is virtually impossible for a person to correctly and completely anticipate all of the arguments that are necessary for a large set of case frames for a domain. Omitting an important argument will result in the failure to recognize role objects in certain syntactic constructions. In practice, people often turn to the corpus to look for argument structures that they might have missed. For example, the UMass/MUC-4 terrorism case frames were developed by applying an initial set of case frames to hundreds of sample texts and looking for places where the case frames failed to recognize desired information. But this approach is extremely time-consuming unless the answers are known in advance (i.e., the information that should have been extracted), which is unrealistic for most applications.

It should be possible, however, to learn case frame structures automatically from a text corpus. Toward this end, we have been developing a corpus-based approach to conceptual case frame acquisition. Our approach builds upon earlier work on corpus-based methods for generating extraction patterns (Riloff, 1996b) and semantic lexicons (Riloff and Shepherd, 1997). Our new system constructs conceptual case frames by learning semantic preferences for extraction patterns and merging syntactically compatible patterns into more complex structures. The resulting case frames can have slots for multiple role objects and each slot has a set of learned selectional restrictions for its role object.

The first section of this paper begins with background about AutoSlog-TS, a corpus-based system for generating extraction patterns automatically, and the extraction patterns that it generates. The following section presents a new corpus-based algorithm that uses the extraction patterns as a building block for constructing conceptual case frame structures. We then show several examples of case frames that were generated automatically using this method. Finally, we present experimental results that compare the performance of the case frames with the extraction patterns. Our results show that the conceptual case frames produce substantially fewer false hits than the extraction patterns.

2 AutoSlog-TS: generating simple extraction patterns

In the past few years, several systems have been developed to generate structures for information extraction automatically. However, these systems usually need special training resources that are expensive to obtain. One of the first such systems was AutoSlog (Riloff, 1993; Riloff, 1996a), which generates extraction patterns from annotated text. The patterns produced by AutoSlog achieved 98% of the performance of hand-crafted extraction patterns, but AutoSlog requires a training corpus that is manually tagged with domain-specific annotations. Another early system, PALKA (Kim and Moldovan, 1993), requires domain-specific frames with keyword lists, CRYSTAL (Soderland et al., 1995) requires an annotated training corpus, RAPIER (Califf and Mooney, 1997) requires filled templates, and LIEP (Huffman, 1996) requires keywords and annotated training examples. PALKA and CRYSTAL also require semantic lexicons, while LIEP uses domain-specific concept recognizers.

AutoSlog-TS (Riloff, 1996b) is a derivative of AutoSlog that was designed to obviate the need for special training data. AutoSlog-TS generates extraction patterns using only a "preclassified" training corpus: one set of texts that are relevant to the domain, and one set of texts that are irrelevant. The texts do not need to be annotated in any way.

AutoSlog-TS generates the same simple extraction patterns that AutoSlog generates. Each pattern is activated by a keyword in a specific linguistic context. For example, one extraction pattern may be triggered by the word "murdered" in passive verb constructions, while a different extraction pattern may be triggered by "murdered" in active verb constructions. Each pattern extracts information from a syntactic constituent in the current clause: the subject, the direct object, or a prepositional phrase.

AutoSlog-TS generates extraction patterns by making two passes over the corpus. In the first pass, AutoSlog-TS uses AutoSlog's heuristics in an exhaustive fashion to generate a set of patterns that collectively extract every noun phrase in the corpus. In the second pass, AutoSlog-TS computes statistics to determine which extraction patterns are most strongly correlated with the relevant training texts. The patterns are ranked so that those most strongly associated with the domain appear at the top. Figure 1 shows the top 20 extraction patterns produced by AutoSlog-TS for the MUC-4 terrorism domain (MUC-4 Proceedings, 1992). The ranked list is then presented to a human to decide which patterns should be kept. For example, the pattern "<subject> exploded" should be retained because it is likely to extract relevant information about bombings. However, the pattern "<subject> said" should be discarded because it is not likely to extract information about terrorism and will probably extract a lot of irrelevant information. The human reviewer assigns a conceptual role to each accepted pattern to characterize its extractions. For example, the pattern "<subject> was murdered" would be assigned the role *victim* for its extractions.

1. <subject> exploded
2. <subject> reported
3. <subject> was killed
4. <subject> located
5. <subject> took_place
6. <subject> was kidnapped
7. <subject> was injured
8. <subject> carried_out
9. caused <direct-obj>
10. <subject> was wounded
11. <subject> caused
12. <subject> occurred
13. claimed <direct-obj>
14. <subject> was murdered
15. murder of <noun-phrase>
16. <subject> claimed responsibility
17. <subject> was reported
18. <subject> said
19. exploded in <noun-phrase>
20. <subject> kidnapped

Figure 1: Top 20 extraction patterns for a terrorism domain

The extraction patterns learned by AutoSlog-TS (and AutoSlog) have two serious limitations. First, each pattern extracts only one item, which causes the output to be artificially fragmented. For example, the sentence "Guerrillas kidnapped the mayor in Bogota" produces three extractions (Guerrillas, the mayor, and Bogota), each in a separate structure. This fragmented representation causes unnecessary work for subsequent components that need to piece the information back together. Second, the patterns do not include semantic constraints so they produce many spurious extractions.[1]

Theoretically, conceptual case frames should overcome both of these limitations. Multi-slot case frames will allow several role objects associated with the same event to be instantiated as part of the same structure. This produces a more coherent representation, which is more natural for subsequent event or discourse processing. Furthermore, if each slot has selectional restrictions associated with its legal role objects, then the case frames should produce fewer false hits (i.e., spurious extractions).

In the next section, we describe a corpus-based algorithm that constructs conceptual case frames empirically by learning semantic preferences for each extraction pattern and using these preferences to assign conceptual roles automatically. (Consequently, the human reviewer no longer needs to assign roles to the extraction patterns manually.) Extraction patterns with compatible syntactic constraints are then merged to produce multi-slot case frames with selectional restrictions. The conceptual case frames should be more reliable at identifying relevant information (our experimental results support this hypothesis), and the case frames can instantiate multiple role objects in a single structure to simplify subsequent discourse processing.

3 Generating conceptual case frames from extraction patterns

The algorithm for building conceptual case frames begins with extraction patterns and a semantic lexicon for the domain. The semantic lexicon is a dictionary of words that belong to relevant semantic categories. We used AutoSlog-TS to generate the extraction patterns and a corpus-based algorithm to generate the semantic lexicon.[2]

The corpus-based algorithm that we used to build the semantic lexicon (Riloff and Shepherd, 1997) requires five "seed words" as input for each semantic category, and produces a ranked list of words that are statistically associated with each category. First, the algorithm looks for all sentences in which a seed word is used as the head noun of a noun phrase. For each such occurrence of a seed word, the algorithm collects a small context window around the seed word. The context window consists of the closest noun to the left of the seed word, and the closest noun to its right. The context windows for all seed words that belong to the same category are then combined, and each word is assigned a category score. The category score is (essentially) the conditional probability that the word appears in a category context. The words are ranked by this score and the top five are dynamically added to the seed word list. This bootstrapping process dynamically grows the seed word list so that each iteration produces a larger category context. After several iterations, the final list of ranked words usually contains many words that belong to the category, especially near the top. The ranked list is presented to a user, who scans down the list and removes any words that do not belong to the category. For more details of this algorithm, see (Riloff and Shepherd, 1997).

A flowchart for the case frame generation process appears in Figure 2. AutoSlog-TS produces a ranked list of extraction patterns and our semantic lexicon generator produces a ranked list of words for each category. Generating these lists is fully automatic, but a human must review them to decide which extraction patterns and category words to keep. This is the only part of the process that involves human interaction.

[1] Semantic constraints could be associated with the conceptual roles assigned by the human reviewer, but our goal is to assign both the conceptual roles and selectional restrictions automatically.

[2] Other methods could be used to generate these items, including the use of existing knowledge bases such as WordNet (Miller, 1990) or Cyc (Lenat et al., 1986) if they have adequate coverage for the domain.

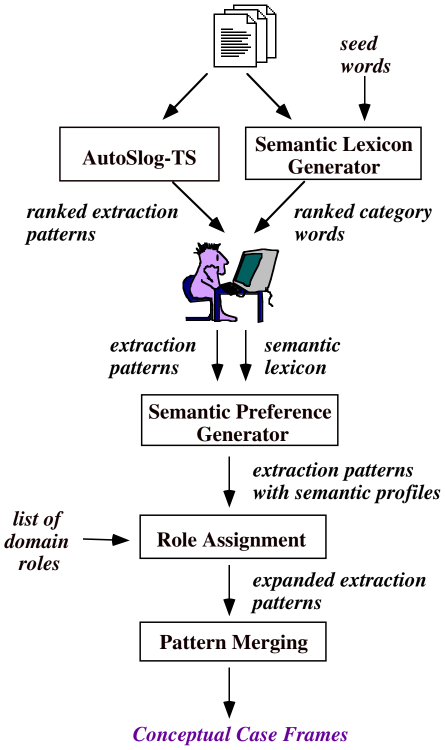

Figure 2: Generating case frames

Next, the extraction patterns are applied to the texts to generate a *semantic profile* for each pattern. The semantic profile shows the semantic categories that were extracted by each pattern, based on the head noun of each extraction. Figure 3 shows the semantic profile for the pattern "attack on <noun-phrase>". **PFreq** is the number of times that the extraction pattern fired, **SFreq** is the number of times that the pattern extracted the given semantic category, and **Prob** is the estimated probability of the pattern extracting the given semantic category (**SFreq/PFreq**). Note that many extractions will not be labeled with any semantic category if the head noun is unknown (i.e., not in the semantic lexicon).

Figure 3 shows that attacks are often carried out on buildings, civilians, dates, government officials, locations, military people, and vehicles. It seems obvious that attacks will occur on people and on physical targets, but a person might not realize that attacks will also occur on dates (e.g., Monday) and on locations (e.g., a neighborhood). This example shows how the corpus-based approach can identify semantic preferences that a person might not anticipate. Also, note that the semantic profile shows no instances of attacks on terrorists or weapons, which makes sense in this domain.

Sem. Category	PFreq	SFreq	Prob
BUILDING	149	15	0.10
CIVILIAN	149	5	0.03
DATE	149	7	0.05
GOVOFFICIAL	149	4	0.03
LOCATION	149	4	0.03
MILITARYPEOPLE	149	13	0.09
TERRORIST	149	0	0.00
VEHICLE	149	4	0.03
WEAPON	149	0	0.00

Figure 3: Semantic profile for "attack on <noun-phrase>"

The semantic profile is used to select semantic preferences that are strong enough to become selectional restrictions. We use the following formula to identify strong semantic preferences:

(SFreq $\geq$ F1) or ((SFreq $\geq$ F2) and (Prob $\geq$ P))

The first test selects semantic categories that are extracted with high frequency, under the assumption that this reflects a real association with the category. The second case selects semantic categories that represent a relatively high percentage of the extractions even though the frequency might be low (e.g., 2 out of 4 extractions). In our experiments, we chose F1=3, F2=2, and P=0.1. We used fairly lenient criteria because (a) patterns can often extract several types of objects that belong to different semantic categories, and (b) many extractions contain unknown words. Also, remember that the semantic lexicon is reliable because it was reviewed by a person, so it is usually meaningful when a pattern extracts a semantic category even once. The thresholds are needed only to eliminate noise, which can be caused by misparsed sentences or polysemous words.

The semantic preferences are used to assign conceptual roles to each extraction pattern. At this point, one additional piece of input is needed: a list of conceptual roles and associated semantic categories for the domain. The conceptual roles identify the types of information that need to be recognized. Figure 4 shows the conceptual roles used for the terrorism domain.

Domain Role	Semantic Categories
Perpetrator	TERRORIST
Target	BUILDING, VEHICLE
Victim	CIVILIAN, GOVOFFICIAL
Location	LOCATION
Instrument	WEAPON
Date	TIME

Figure 4: Conceptual roles for terrorism

Each extraction pattern is expanded to include a set of conceptual roles based on its semantic preferences. These conceptual roles are assigned automatically based on a pattern's semantic profile. This process eliminates the need for a human to assign roles to the extraction patterns by hand, as had been necessary when using AutoSlog or AutoSlog-TS by themselves.

For example, the pattern "machinegunned <direct-obj>" had strong semantic preferences for BUILDING, CIVILIAN, LOCATION, and VEHICLE, so it was expanded to have three conceptual roles with four selectional restrictions. The expanded extraction pattern for "machinegunned <direct-obj>" is:

"machinegunned <direct-obj>" →

Victim	CIVILIAN
Target	BUILDING VEHICLE
Location	LOCATION

Only semantic categories that were associated with a pattern are included as selectional restrictions. For example, the GOVOFFICIAL category also represents possible terrorism victims, but it was not strongly associated with the pattern. Our rationale is that an individual pattern may have a strong preference for only a subset of the categories that can be associated with a role. For example, the pattern "<subject> was ambushed" showed a preference for VEHICLE extractions but not BUILDING extractions, which makes sense because it is hard to imagine ambushing a building. Including only VEHICLE as its selectional restriction for targets might help eliminate incorrect building extractions. One could argue that this pattern is not likely to find building extractions anyway so the selectional restriction will not matter, but the selectional restriction might help filter out incorrect extractions due to misparses or metaphor (e.g., "The White House was ambushed by reporters."). Ultimately, it is an empirical question whether it is better to include all of the semantic categories associated with a conceptual role or not.

Finally, we merge the expanded extraction patterns into multi-slot case frames. All extraction patterns that share the same trigger word and compatible syntactic constraints are merged into a single structure. For example, we would merge all patterns triggered by a specific verb in its passive voice. For example, the patterns "<subject> was kidnapped", "was kidnapped by <noun-phrase>", and "was kidnapped in <noun-phrase>" would be merged into a single case frame. Similarly, we would merge all patterns triggered by a specific verb in its active voice. For example, we would merge patterns for the active form of "destroyed" that extract the subject of "destroyed", its direct object, and any prepositional phrases that are associated with it. We also merge syntactically compatible patterns that are triggered by the same noun (e.g., "assassination") or by the same infinitive verb structure (e.g., "to kill"). When we merge extraction patterns into a case frame, all of the slots are simply unioned together.

4 Examples

In this section, we show several examples of case frames that were generated automatically by our system. Figure 5 shows a simple case frame triggered by active forms of the verb "ambushed". The subject is extracted as a *perpetrator* and has a selectional restriction of TERRORIST. The direct object is extracted as a *target* and has a selectional restriction of VEHICLE. Note that the case frame does not contain a *victim* slot, even though it is theoretically possible to ambush people. During training, the "ambushed <direct-obj>" pattern extracted 13 people, 11 of whom were recognized as MILITARYPEOPLE. Since our domain roles only list civilians and government officials as legitimate terrorism victims[3], a victim slot was not created. This example shows how the case frames are tailored for the domain empirically.

Caseframe:	(active_verb ambushed)	
perpetrator	subject	TERRORIST
target	direct-obj	VEHICLE

Figure 5: Case frame for active forms of "ambushed"

Figure 6 shows a case frame triggered by active forms of "blew_up".[4] This case frame extracts information from an entire sentence into a single structure. The subject (*perpetrator*), direct object (*target*), and a prepositional phrase (in *location*) will all be extracted together.

Caseframe:	(active_verb blew_up)	
perpetrator	subject	TERRORIST
target	direct-obj	BUILDING VEHICLE
location	pp(in)	LOCATION

Figure 6: Case frame for active forms of "blew_up"

The case frame in Figure 7 illustrates how a semantic category can show up in multiple places. This case frame will handle phrases like "the guerrillas detonated a bomb", as well as "the bomb detonated". Both constructions are very common in the training corpus so the system added slots for both possibilities. It would be easy for a human to overlook some of these variations when creating case frames by hand.

The case frame in Figure 8 is activated by the noun "attack" and includes slots for a variety of prepositional phrases. The same preposition can recognize different types of information (e.g., "on" can recognize *targets*, *victims*, *locations*, and *dates*). And the same role can be filled by different prepositions

[3] Events involving military victims were classified as military incidents, not terrorism, according to the MUC-4 guidelines.

[4] Underscored words represent lexicalized expressions in our phrasal lexicon.

Caseframe:	(active_verb detonated)	
perpetrator	subject	TERRORIST
instrument	subject	WEAPON
instrument	direct-obj	WEAPON

Figure 7: Case frame for active forms of "detonated"

(e.g., *targets* can be extracted from "on", "against", or "at"). This example again shows the power of corpus-based methods to identify common constructions empirically. Anticipating all of these prepositional arguments would be difficult for a person.

Caseframe:	(noun attack)	
target	pp(on)	BUILDING VEHICLE
victim	pp(on)	CIVILIAN GOVOFFICIAL
location	pp(on)	LOCATION
date	pp(on)	TIME
target	pp(against)	BUILDING VEHICLE
victim	pp(against)	CIVILIAN
target	pp(at)	BUILDING
location	pp(at)	LOCATION

Figure 8: Case frame for noun forms of "attack"

A disadvantage of this automated method is that inappropriate slots sometimes end up in the case frames. For example, Figure 9 shows a case frame that is activated by passive forms of the verb "killed". Some of the slots are correct: the subject is assigned to the *victim* slot and objects of the preposition "by" are assigned to the *perpetrator* and *instrument* slots. However, the remaining slots do not make sense. The *location* slot is the result of polysemy; many person names are also location names, such as "Flores". The *date* slot was produced by incorrect parses of date expressions. The *perpetrator* (subject) and *victim* (pp (by)) slots were caused by incorrect role assignments. The list of domain roles assumes that terrorists are always perpetrators and civilians are always victims, but of course this is not true. Terrorists can be killed and civilians can be killers.

Caseframe:	(passive_verb killed)	
victim	subject	CIVILIAN GOVOFFICIAL
perpetrator	subject	TERRORIST
location	subject	LOCATION
date	subject	TIME
perpetrator	pp(by)	TERRORIST
victim	pp(by)	CIVILIAN
instrument	pp(by)	WEAPON

Figure 9: Case frame for passive forms of "killed"

The previous example illustrates some of the problems that can occur when generating case frames automatically. Currently, we are assuming that each semantic category will be uniquely associated with a conceptual role, which may be an unrealistic assumption for some domains. One avenue for future work is to develop more sophisticated methods for mapping semantic preferences to conceptual roles. One could also have a human review the case frames and manually remove inappropriate slots. For now, we chose to avoid additional human interaction and used the case frames exactly as they were generated.

5 Evaluation

The purpose of the selectional restrictions is to constrain the types of information that can be instantiated by each slot. Consequently, we hoped that the case frames would be more reliably instantiated than the extraction patterns, thereby producing fewer false hits. To evaluate the case frames, we used the same corpus and evaluation metrics as previous experiments with AutoSlog and AutoSlog-TS (Riloff, 1996b) so that we can draw comparisons between them. For training, we used the 1500 MUC-4 development texts to generate the extraction patterns and the semantic lexicon. AutoSlog-TS generated 44,013 extraction patterns in its first pass. After discarding the patterns that occurred only once, the remaining 11,517 patterns were applied to the corpus for the second pass and ranked for manual review. We reviewed the top 2168 patterns[5] and kept 306 extraction patterns for the final dictionary.

We built a semantic lexicon for nine categories associated with terrorism: BUILDING, CIVILIAN, GOVOFFICIAL, MILITARYPEOPLE, LOCATION, TERRORIST, DATE, VEHICLE, WEAPON. We reviewed the top 500 words for each category. It takes about 30 minutes to review a category assuming that the reviewer is familiar with the domain. Our final semantic dictionary contained 494 words. In total, the review process required approximately 6 person-hours: 1.5 hours to review the extraction patterns plus 4.5 hours to review the words for 9 semantic categories. From the extraction patterns and semantic lexicon, our system generated 137 conceptual case frames.

One important question is how to deal with unknown words during extraction. This is especially important in the terrorism domain because many of the extracted items are proper names, which cannot be expected to be in the semantic lexicon. We allowed unknown words to fill all eligible slots and then used a precedence scheme so that each item was instantiated by only one slot. Precedence was based on the order of the roles shown in Figure 4. This is not a very satisfying solution and one of the weaknesses of our current approach. Handling unknown words more intelligently is an important direction for future research.

We compared AutoSlog-TS' extraction patterns

[5]We decided to review the top 2000 but continued down the list until there were no more ties.

Slot	cor	mis	mlb	dup	spu	R	P
Perp	25	31	10	18	84	.45	.31
Victim	44	23	16	24	62	.66	.47
Target	31	22	17	23	66	.58	.39
Instr	16	15	7	17	23	.52	.52
Total	116	91	50	82	235	.56	.41

Table 1: AutoSlog-TS results

Slot	cor	mis	mlb	dup	spu	R	P
Perp	26	30	4	17	71	.46	.36
Victim	38	28	24	12	26	.58	.50
Target	28	25	3	29	48	.53	.53
Instr	17	14	2	19	8	.55	.78
Total	109	97	33	77	153	.53	.50

Table 2: Case frame results

with the case frames using 100 blind texts[6] from the MUC-4 test set. The MUC-4 answer keys were used to score the output. Each extracted item was scored as either *correct*, *mislabeled*, *duplicate*, or *spurious*. An item was *correct* if it matched against the answer keys. An item was *mislabeled* if it matched against the answer keys but was extracted as the wrong type of object (e.g., if a victim was extracted as a perpetrator). An item was a *duplicate* if it was coreferent with an item in the answer keys. Correct items extracted more than once were scored as duplicates, as well as correct but underspecified extractions such as "Kennedy" instead of "John F. Kennedy".[7] An item was *spurious* if it did not appear in the answer keys. All items extracted from irrelevant texts were spurious. Finally, items in the answer keys that were not extracted were counted as *missing*. *Correct* + *missing* equals the total number of items in the answer keys.[8]

Table 1 shows the results[9] for AutoSlog-TS' extraction patterns, and Table 2 shows the results for the case frames. We computed **Recall (R)** as *correct* / (*correct* + *missing*), and **Precision (P)** as (*correct* + *duplicate*) / (*correct* + *duplicate* + *mislabeled* + *spurious*). The extraction patterns and case frames achieved similar recall results, although the case frames missed seven correct extractions. However the case frames produced substantially fewer false hits, producing 82 fewer spurious extractions.

Note that perpetrators exhibited by far the lowest precision. The reason is that the perpetrator slot received highest precedence among competing slots for unknown words. Changing the precedence scheme produces a bubble effect where many incorrect extractions shift to the primary default category. The case frames therefore have the potential for even higher precision if the unknown words are handled better. Expanding the semantic lexicon is one option, and additional work may suggest ways to choose slots for unknown words more intelligently.

6 Conclusions

We have shown that conceptual case frames can be generated automatically using unannotated text as input, coupled with a few hours of manual review. Our results for the terrorism domain show that the case frames achieve similar recall levels as the extraction patterns, but with substantially fewer false hits. Our results are not directly comparable to the MUC-4 results because the MUC-4 systems contained additional components, such as domain-specific discourse analyzers that resolved coreferent noun phrases, merged event descriptions, and filtered out irrelevant information. The work presented here only addresses the initial stage of information extraction. However, in previous work we showed that AutoSlog-TS achieved performance comparable to AutoSlog (Riloff, 1996b), which performed very well in the MUC-4 evaluation (Lehnert et al., 1992b). Since the conceptual case frames achieved comparable recall and higher precision than AutoSlog-TS' extraction patterns, our results suggest that the case frames performed well relative to previous work on this domain.

Several other systems learn extraction patterns that can also be viewed as conceptual case frames with selectional restrictions (e.g., PALKA (Kim and Moldovan, 1993) and CRYSTAL (Soderland et al., 1995)). The case frames learned by our system are not necessarily more powerful then those generated by other systems. The advantage of our approach is that it requires no special training resources. Our technique requires only preclassified training texts and a few hours of manual filtering to build the intermediate dictionaries. Given preclassified texts, it is possible to build a dictionary of conceptual case frames for a new domain in one day.

Another advantage of our approach is its highly empirical nature; a corpus often reveals important patterns in a domain that are not necessarily intuitive to people. By using corpus-based methods to generate all of the intermediate dictionaries and

[6] 25 relevant texts and 25 irrelevant texts from each of the TST3 and TST4 test sets.

[7] The rationale for scoring coreferent phrases as duplicates instead of spurious is that the extraction pattern or case frame was instantiated with a reference to the correct answer. In other words, the pattern (or case frame) did the right thing. Resolving coreferent phrases to produce the best answer is a problem for subsequent discourse analysis, which is not addressed by the work presented here.

[8] A caveat is that the MUC-4 answer keys contain some "optional" answers. We scored these as correct if they were extracted but they were never scored as missing, which is how the "optional" items were scored in MUC-4. Note that the number of possible extractions can vary depending on the output of the system.

[9] We reimplemented AutoSlog-TS to use a different sentence analyzer, so these results are slightly different from those reported in (Riloff, 1996b).

the final case frame structures, the most important words, role assignments, and semantic preferences are less likely to be missed. Our empirical approach aims to exploit the text corpus to automatically acquire the syntactic and semantic role assignments that are necessary to achieve good performance in the domain.

7 Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. IRI-9509820 and IRI-9704240.

References

M. E. Califf and R. J. Mooney. 1997. Relational Learning of Pattern-Match Rules for Information Extraction. In *Proceedings of the ACL Workshop on Natural Language Learning*, pages 9–15.

S. Huffman. 1996. Learning information extraction patterns from examples. In Stefan Wermter, Ellen Riloff, and Gabriele Scheler, editors, *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, pages 246–260. Springer-Verlag, Berlin.

J. Kim and D. Moldovan. 1993. Acquisition of Semantic Patterns for Information Extraction from Corpora. In *Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications*, pages 171–176, Los Alamitos, CA. IEEE Computer Society Press.

W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. 1992a. University of Massachusetts: Description of the CIRCUS System as Used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 282–288, San Mateo, CA. Morgan Kaufmann.

W. Lehnert, C. Cardie, D. Fisher, J. McCarthy, E. Riloff, and S. Soderland. 1992b. University of Massachusetts: MUC-4 Test Results and Analysis. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 151–158, San Mateo, CA. Morgan Kaufmann.

D. B. Lenat, M. Prakash, and M. Shepherd. 1986. CYC: Using Common Sense Knowledge to Overcome Brittleness and Knowledge-Acquisition Bottlenecks. *AI Magazine*, 6:65–85.

G. Miller. 1990. Wordnet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4).

MUC-4 Proceedings. 1992. *Proceedings of the Fourth Message Understanding Conference (MUC-4)*. Morgan Kaufmann, San Mateo, CA.

E. Riloff and J. Shepherd. 1997. A Corpus-Based Approach for Building Semantic Lexicons. In *Proceedings of the Second Conference on Empirical Methods in Natural Language Processing*, pages 117–124.

E. Riloff. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 811–816. AAAI Press/The MIT Press.

E. Riloff. 1996a. An Empirical Study of Automated Dictionary Construction for Information Extraction in Three Domains. *Artificial Intelligence*, 85:101–134.

E. Riloff. 1996b. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, pages 1044–1049. The AAAI Press/MIT Press.

S. Soderland, D. Fisher, J. Aseltine, and W. Lehnert. 1995. CRYSTAL: Inducing a conceptual dictionary. In *Proceedings of the Fourteenth International Joint Conference on Artificial Intelligence*, pages 1314–1319.