Effective Information Extraction with Semantic Affinity Patterns and Relevant Regions

Siddharth Patwardhan and **Ellen Riloff**
School of Computing
University of Utah
Salt Lake City, UT 84112
{sidd,riloff}@cs.utah.edu

Abstract

We present an information extraction system that decouples the tasks of finding relevant regions of text and applying extraction patterns. We create a self-trained relevant sentence classifier to identify relevant regions, and use a *semantic affinity* measure to automatically learn domain-relevant extraction patterns. We then distinguish *primary* patterns from *secondary* patterns and apply the patterns selectively in the relevant regions. The resulting IE system achieves good performance on the MUC-4 terrorism corpus and ProMed disease outbreak stories. This approach requires only a few seed extraction patterns and a collection of relevant and irrelevant documents for training.

1 Introduction

Many information extraction (IE) systems rely on rules or patterns to extract words and phrases based on their surrounding context (e.g., (Soderland et al., 1995; Riloff, 1996; Califf and Mooney, 1999; Yangarber et al., 2000)). For example, a pattern like *"<subject> was assassinated"* can reliably identify a victim of a murder event. Classification-based IE systems (e.g., (Freitag, 1998; Freitag and McCallum, 2000; Chieu et al., 2003)) also generally decide whether to extract words based on properties of the words themselves as well as properties associated with their surrounding context.

In this research, we propose an alternative approach to IE that decouples the tasks of finding a relevant region of text and finding a desired extraction. In a typical pattern-based IE system, the extraction patterns perform two tasks: (a) they recognize that a relevant incident has occurred, and (b) they identify and extract some information about that event. In contrast, our approach first identifies relevant regions of a document that describes relevant events, and then applies extraction patterns only in these relevant regions.

This decoupled approach to IE has several potential advantages. First, even seemingly good patterns can produce false hits due to metaphor and idiomatic expressions. However, by restricting their use to relevant regions of text, we could avoid such false positives. For example, *"John Kerry attacked George Bush"* is a metaphorical description of a verbal tirade, but could be easily mistaken for a physical attack. Second, IE systems are prone to errors of omission when relevant information is not explicitly linked to an event. For instance, a phrase like *"the gun was found..."* does not directly state that the the gun was used in a terrorist attack. But if the gun is mentioned in a region that clearly describes a terrorist attack, then it can be reasonably inferred to have been used in the attack. Third, if the IE patterns are restricted to areas of text that are known to be relevant, then it may suffice to use relatively general patterns, which may be easier to learn or acquire.

Our approach begins with a relevant sentence classifier that is trained using only a few seed patterns and a set of relevant and irrelevant documents (but no sentence-level annotations) for the domain of interest. The classifier is then responsible for identifying sentences that are relevant to the IE task. Next, we learn "semantically appropriate" extraction pat-

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 717–727, Prague, June 2007.

terns by evaluating candidate patterns using a *semantic affinity* metric. We then separate the patterns into *primary* and *secondary* patterns, and apply them selectively to sentences based on the relevance judgments produced by the classifier. We evaluate our IE system on two data sets: the MUC-4 IE terrorism corpus and ProMed disease outbreak articles. Our results show that this approach works well, often outperforming the AutoSlog-TS IE system which benefits from human review.

2 Motivation and Related Work

Our research focuses on event-oriented information extraction (IE), where the goal of the IE system is to extract facts associated with domain-specific events from unstructured text. Many different approaches to information extraction have been developed, but generally speaking they fall into two categories: classifier-based approaches and rule/pattern-based approaches.

Classifier-based IE systems use machine learning techniques to train a classifier that sequentially processes a document looking for words to be extracted. Examples of classifier-based IE systems are SRV (Freitag, 1998), HMM approaches (Freitag and McCallum, 2000), ALICE (Chieu et al., 2003), and Relational Markov Networks (Bunescu and Mooney, 2004). The classifier typically decides whether a word should be extracted by considering features associated with that word as well as features of the words around it.

Another common approach to information extraction uses a set of explicit patterns or rules to find relevant information. Some older systems relied on hand-crafted patterns, while more recent systems learn them automatically or semi-automatically. Examples of rule/pattern-based approaches to information extraction are FASTUS (Hobbs et al., 1997), PALKA (Kim and Moldovan, 1993), LIEP (Huffman, 1996), CRYSTAL (Soderland et al., 1995), AutoSlog/AutoSlog-TS (Riloff, 1993; Riloff, 1996), RAPIER (Califf and Mooney, 1999), WHISK (Soderland, 1999), ExDisco (Yangarber et al., 2000), SNOWBALL (Agichtein and Gravano, 2000), $(LP)^2$ (Ciravegna, 2001), subtree patterns (Sudo et al., 2003), predicate-argument rules (Yakushiji et al., 2006) and KnowItAll (Popescu et al., 2004).

One commonality behind all of these approaches is that they simultaneously decide whether a context is relevant and whether a word or phrase is a desirable extraction. Classifier-based systems rely on features that consider both the word and its surrounding context, and rule/pattern-based systems typically use patterns or rules that match both the words around a candidate extraction and (sometimes) properties of the candidate extraction itself.

There is a simplicity and elegance to having a single model that handles both of these problems at the same time, but we hypothesized that there may be benefits to decoupling these tasks. We investigate an alternative approach that involves two passes over a document. In the first pass, we apply a *relevant region identifier* to identify regions of the text that appear to be especially relevant to the domain of interest. In the second pass, we apply extraction patterns inside the relevant regions. We hypothesize three possible benefits of this decoupled approach.

First, if a system is certain that a region is relevant, then it can be more aggressive about searching for extractions. For example, consider the domain of terrorist event reports, where a goal is to identify the weapons that were used. Existing systems generally require rules/patterns to recognize a context in which a weapon is explicitly linked to an event or its consequences (e.g., *"attack with <np>"*, or *"<np> caused damage"*). However, weapons are not always directly linked to an event in text, but they may be inferred through context. For instance, an article may mention that a weapon was "found" or "used" without explicitly stating that it was involved in a terrorist event. However, if we know in advance that we are in a relevant context, then we can reliably infer that the weapon was, most likely, used in the event.

Second, some patterns may seem to be relevant locally, but they can be deemed irrelevant when the global context is considered. For example, consider these sentences from the MUC-4 terrorism corpus:

> *D'Aubuisson unleashed harsh attacks on Duarte ...*
> *Other brave minds that advocated reform had been killed before in that struggle.*

Locally, patterns such as *"<subject> unleashed* *attacks"* and *"<subject> had been killed"* seem likely to identify the perpetrators and victims of a physical attack. But when read in the full context of these sentences, it becomes clear that they are not related to a specific physical attack.

Third, decoupling these tasks may simplify the learning process. Identifying relevant regions amounts to a text classification task, albeit the goal is to identify not just relevant documents, but relevant sub-regions of documents. Within a relevant region the patterns may not need to be as discriminating. So a more general learning approach may suffice.

In this paper, we describe an IE system that consists of two decoupled modules for relevant sentence identification and extraction pattern learning. In Section 3, we describe the self-trained sentence classifier, which requires only a few seed patterns and relevant and irrelevant documents for training. Section 4 describes the extraction pattern learning module, which identifies semantically appropriate patterns for the IE system using a *semantic affinity* measure. Section 5 explains how we distinguish Primary patterns from Secondary patterns. Section 6 presents experimental results on two domains. Finally, Section 7 lists our conclusions and future work.

3 A Self-Trained Relevant Sentence Classifier

Our hypothesis is that if a system can reliably identify relevant regions of text, then extracting information only from these relevant regions can improve IE performance. There are many possible definitions for *relevant region* (e.g., Salton et al. (1993), Callan (1994)), and exploring the range of possibilities is an interesting avenue for future work. For our initial investigations of this idea, we begin by simply defining a sentence as our region size. This has the advantage of being an easy boundary line to draw (i.e., it is relatively easy to identify sentence boundaries) and it is a small region size yet includes more context than most current IE systems do[1].

Our goal is to create a classifier that can determine whether a sentence contains information that should be extracted. Furthermore, we wanted to create a classifier that does not depend on manually annotated sentence data so that our system can be easily ported across domains. Therefore, we devised a method to self-train a classifier using a training set of relevant and irrelevant documents for the domain, and a few seed patterns as input. However, this results in an asymmetry in the training set. By definition, if a document is irrelevant to the IE task, then it cannot contain any relevant information. Consequently, *all sentences in an irrelevant document must be irrelevant*, so these sentences form our initial *irrelevant sentences pool*. In contrast, if a document is relevant to the IE task, then there must be at least one sentence that contains relevant information. However, most documents contain a mix of both relevant and irrelevant sentences. Therefore, the sentences from the relevant documents form our *unlabeled sentences pool*.

Figure 1 shows the self-training procedure, which begins with a handful of *seed patterns* to initiate the learning process. The seed patterns should be able to reliably identify some information that is relevant to the IE task. For instance, to build an IE system for terrorist incident reports, we used seed patterns such as *"<subject> was kidnapped"* and *"assassination of <np>"*. The patterns serve as a simple pattern-based classifier to automatically identify some relevant sentences. In *iteration 0* of the self-training loop (shown as dotted lines in Figure 1), the pattern-based classifier is applied to the unlabeled sentences to automatically label some of them as relevant.

Next, an SVM (Vapnik, 1995) classifier[2] is trained using these relevant sentences and an equal number of irrelevant sentences randomly drawn from the irrelevant sentences pool. We artificially created a balanced training set because the set of irrelevant sentences is initially much larger than the set of relevant sentences, and we want the classifier to learn how to identify new relevant sentences. The feature set consists of all unigrams that appear in the training set. The SVM is trained using a linear kernel with the default parameter settings. In a self-training loop, the classifier is then applied to the unlabeled sentences, and all sentences that it classifies as relevant are added to the relevant sentences pool. The classifier is then retrained with all of the

[1]Most IE systems only consider a context window consisting of a few words or phrases on either side of a potential extraction.

[2]We used the freely available SVM^{light} (Joachims, 1998) implementation: http://svmlight.joachims.org

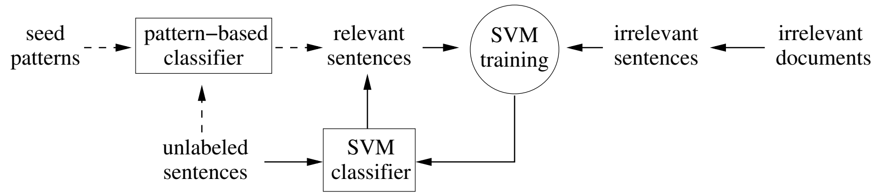

Figure 1: The Training Process to Create a Relevant Sentence Classifier

relevant sentences and an equal number of irrelevant sentences, and the process repeats. We ran this self-training procedure for three iterations and then used the resulting classifier as our *relevant sentence classifier* in the IE experiments described in Section 6.3.

4 Learning Semantic Affinity-based Extraction Patterns

One motivation for creating a relevant region classifier is to reduce the responsibilities of the extraction patterns. Once we know that we are in a domain-relevant area of text, patterns that simply identify words and phrases belonging to a relevant semantic class may be sufficient. In this section, we describe a method to automatically identify semantically appropriate extraction patterns for use with the sentence classifier.

In previous work (Patwardhan and Riloff, 2006), we introduced a metric called *semantic affinity* which was used to automatically assign event roles to extraction patterns. Semantic affinity measures the tendency of a pattern to extract noun phrases that belong to a specific set of semantic categories. To use this metric for information extraction, a mapping must be defined between semantic categories and the event roles that are relevant to the IE task. For example, one role in the terrorism domain is *physical target*, which refers to physical objects that are the target of an attack. Most physical targets fall into one of two general semantic categories: BUILDING or VEHICLE. Consequently, we define the mapping "Target → BUILDING, VEHICLE". Similarly, we might define the mapping "Victim → HUMAN, ANIMAL, PLANT" to characterize possible victims of disease outbreaks. Each semantic category must be mapped to a single event role. This is a limitation of our approach for domains where multiple roles can be filled by the same class of fillers. However, sometimes a general semantic class can be partitioned into subclasses that are associated with different roles. For example, in the terrorism domain, both perpetrators and victims belong to the general semantic class HUMAN. But we used the subclasses TERRORIST-HUMAN, which represents likely perpetrator words (e.g., "terrorist", "guerrilla", and "gunman") and CIVILIAN-HUMAN, which represents ordinary people (e.g., "photographer","rancher", and "tourist"), in order to generate different semantic affinity estimates for the perpetrator and victim roles.

To determine the semantic category of a noun, we use the Sundance parser (Riloff and Phillips, 2004), which contains a dictionary of words that have semantic category labels. Alternatively, a resource such as WordNet (Fellbaum, 1998) could be used to obtain this information. All semantic categories that cannot be mapped to a relevant event role are mapped to a special Other role.

To estimate the semantic affinity of a pattern p for an event role r_k, the system computes $\mathrm{f}(p, r_k)$, which is the number of pattern p's extractions that have a head noun belonging to a semantic category mapped to r_k. These frequency counts are obtained by applying each pattern to the training corpus and collecting its extractions. The *semantic affinity* of a pattern p with respect to an event role r_k is formally defined as:

$$\mathrm{sem\_aff}(p, r_k) = \frac{\mathrm{f}(p, r_k)}{\sum_{i=1}^{|R|} \mathrm{f}(p, r_i)} \log_2 \mathrm{f}(p, r_k) \quad (1)$$

where R is the set of event roles $\{r_1, r_2, \ldots, r_{|R|}\}$. Semantic affinity is essentially the probability that a phrase extracted by pattern p will be a semantically appropriate filler for role r_k, weighted by the log of the frequency.[3] Note that it is possible for a

[3] This formula is very similar to pattern ranking metrics used by previous IE systems (Riloff, 1996; Yangarber et al., 2000), although not for semantics.

pattern to have a semantic affinity for multiple event roles. For instance, a terrorism pattern like *"attack on <np>"* may have a semantic affinity for both Targets and Victims.

To generate extraction patterns for an IE task, we first apply the AutoSlog (Riloff, 1993) extraction pattern generator to the training corpus exhaustively, so that it literally generates a pattern to extract every noun phrase in the corpus. Then for each event role, we rank the patterns based on their semantic affinity for that role.

Figure 2 shows the 10 patterns with the highest semantic affinity scores for 4 event roles. In the terrorism domain, we show patterns that extract *weapons* and *perpetrator organizations* (PerpOrg). In the disease outbreaks domain, we show patterns that extract *diseases* and *victims*. The patterns rely on shallow parsing, syntactic role assignment (e.g., subject (*subject*) and direct object (*dobj*) identification), and active/passive voice recognition, but they are shown here in a simplified form for readability. The portion in brackets (between < and >) is extracted, and the other words must match the surrounding context. In some cases, all of the matched words are extracted (e.g., "<# birds>"). Most of the highest-ranked victim patterns recognize noun phrases that refer to people or animals because they are common in the disease outbreak stories and these patterns do not extract information that is associated with any competing event roles.

Top Terrorism Patterns	
Weapon	PerpOrg
<subject> exploded	<subject> claimed
planted <dobj>	panama from <np>
fired <dobj>	<np> claimed responsibility
<subject> was planted	command of <np>
explosion of <np>	wing of <np>
<subject> was detonated	kidnapped by <np>
<subject> was set off	guerillas of <np>
set off <dobj>	<subject> operating
hurled <dobj>	kingpins of <np>
<subject> was placed	attacks by <np>

Top Disease Outbreak Patterns	
Disease	Victim
cases of <np>	<# people>
spread of <np>	<# cases>
outbreak of <np>	<# birds>
<#th outbreak>	<# animals>
<# outbreaks>	<subject> died
case of <np>	<# crows>
contracted <dobj>	<subject> know
outbreaks of <np>	<# pigs>
<# viruses>	<# cattle>
spread of <np>	<# sheep>

Figure 2: Top-Ranked Extraction Patterns

5 Distinguishing Primary and Secondary Patterns

So far, our goal has been to find relevant areas of text, and then apply semantically appropriate patterns in those regions. Our expectation was that fairly general, semantically appropriate patterns could be effective if their range is restricted to regions that are known to be relevant. If our relevant sentence classifier was perfect, then performing IE only on relevant regions would be ideal. However, identifying relevant regions is a difficult problem in its own right, and our relevant sentence classifier is far from perfect.

Consequently, one limitation of our proposed approach is that no IE would be performed in sentences that are not deemed to be relevant by the classifier, and this could negatively affect recall. We addressed this issue by allowing reliable patterns to be applied to all sentences in the text, irrespective of the output of the sentence classifier. For example, the pattern *"<subject> was assassinated"* is a clear indicator of a murder event, and does not need to be restricted by the sentence classifier[4]. We will refer to such reliable patterns as *Primary Patterns*. In contrast, patterns that are not necessarily reliable and need to be restricted to relevant regions will be called *Secondary Patterns*.

To automatically distinguish Primary Patterns from Secondary Patterns, we compute the conditional probability of a pattern p being relevant, $\Pr(relevant \mid p)$, based on the relevant and irrelevant documents in our training set. We then define an upper conditional probability threshold θ_u to separate Primary patterns from Secondary Patterns. If a pattern has a high correlation with relevant documents, then our assumption is that it is generally a reliable pattern that is not likely to occur in irrelevant contexts.

On the flip side, we can also use this conditional probability to weed out patterns that rarely

[4]In other words, if such a pattern matches a sentence that is classified as irrelevant, then the classifier is probably incorrect.

appear in relevant documents. Such patterns (e.g., "*<subject> held*", "*<subject> saw*", etc.) could potentially have a high semantic affinity for one of the semantic categories, but they are not likely to be useful if they mainly occur in irrelevant documents. As a result, we also define a lower conditional probability threshold θ_l that identifies irrelevant extraction patterns.

The two thresholds θ_u and θ_l are used with semantic affinity to identify the most appropriate Primary and Secondary patterns for the task. This is done by first removing from our extraction pattern collection all patterns with probability less than θ_l. For each event role, we then sort the remaining patterns based on their semantic affinity score for that role and select the top N patterns. Next, we use the θ_u probability threshold to separate these N patterns into two subsets. Patterns with a probability above θ_u are considered to be Primary patterns for that role, and those below become the Secondary patterns.

6 Experiments and Results

6.1 Data Sets

We evaluated the performance of our IE system on two data sets: the MUC-4 terrorism corpus (Sundheim, 1992), and a ProMed disease outbreaks corpus. The MUC-4 IE task is to extract information about Latin American terrorist events. We focused our analysis on five MUC-4 string roles: *perpetrator individuals*, *perpetrator organizations*, *physical targets*, *victims*, and *weapons*. The disease outbreaks corpus consists of electronic reports about disease outbreak events. For this domain we focused on two string roles: *diseases* and *victims*[5].

The MUC-4 data set consists of 1700 documents, divided into 1300 development (DEV) texts, and four test sets of 100 texts each (TST1, TST2, TST3, and TST4). We used 1300 texts (DEV) as our training set, 200 texts (TST1+TST2) for tuning, and 200 texts (TST3+TST4) as a test set. All 1700 documents have answer key templates. For the training set, we used the answer keys to separate the documents into relevant and irrelevant subsets. Any document containing at least one relevant event was considered relevant.

[5]The "victims" can be people, animals, or plants that are affected by a disease.

For the disease outbreak domain the data set was collected from ProMed-mail[6], an open-source, global electronic reporting system for outbreaks of infectious diseases. We collected thousands of ProMed reports and created answer key templates for 245 randomly selected articles. We used 125 as a tuning set, and 120 as the test set. We used 2000 different documents as the relevant documents for training. Most of the ProMed articles contain email headers, footers, citations, and other snippets of non-narrative text, so we wrote a "zoner" program[7] to automatically strip off some of this extraneous information.

To obtain irrelevant documents, we collected 4000 biomedical abstracts from PubMed[8], a free archive of biomedical literature. We collected twice as many irrelevant documents because the PubMed articles are roughly half the size of the ProMed articles, on average. To ensure that the PubMed articles were truly irrelevant (i.e. did not contain any disease outbreak reports) we used specific queries to exclude disease outbreak abstracts.

The complete IE task involves the creation of answer key templates, one template per incident[9]. Template generation is a complex process, requiring coreference resolution and discourse analysis to determine how many incidents were reported and which facts belong with each incident. Our work focuses on extraction pattern learning and not template generation, so we evaluated our systems directly on the extractions themselves, before template generation would take place. This approach directly measures how accurately the patterns find relevant information, without confounding factors from the template generation process. For example, if a coreference resolver incorrectly decides that two extractions are coreferent and merges them, then only one extraction would be scored. We used a *head noun* scoring scheme, where an extraction is considered to be correct if its head noun matches the head noun in the answer key[10]. Also, pronouns were discarded from both the system responses and the answer keys since no coreference resolution is done. Duplicate

[6]http://www.promedmail.org

[7]The term *zoner* was introduced by Yangarber et al. (2002).

[8]http://www.pubmedcentral.nih.gov

[9]Many of the stories have multiple incidents per article.

[10]For example, "armed men" will match "5 armed men".

extractions (e.g., the same string extracted by different patterns) were conflated before being scored, so they count as just one hit or one miss.

6.2 Relevant Sentence Classifier Results

First, we evaluated the performance of the relevant sentence classifier described in Section 3. We automatically generated seed patterns from the training texts. AutoSlog (Riloff, 1993) was used to generate all extraction patterns that appear in the training documents, and only those patterns with frequency > 50 were kept. These were then ranked by $\Pr(relevant \mid p)$, and the top 20 patterns were chosen as seeds. In the disease outbreak domain, 54 patterns had a frequency > 50 and probability of 1.0. We wanted to use the same number of seeds in both domains for consistency, so we manually reviewed them and used the 20 most domain-specific patterns as seeds.

Due to the greater stylistic differences between the relevant and irrelevant documents in the disease outbreak domain (since they were gathered from different sources), we decided to make the classifier for that domain more conservative in classifying documents as relevant. To do this we used the prediction scores output by the SVM as a measure of confidence in the classification. These scores are essentially the distance of the test examples from the support vectors of the SVM. For the disease outbreaks domain we used a cutoff of 1.0 and in the terrorism domain we used the default of 0.

Since we do not have sentence annotated data, there is no direct way to evaluate the classifiers. However, we did an indirect evaluation by using the answer keys from the tuning set. If a sentence in a tuning document contained a string that occurred in the corresponding answer key template, then we considered that sentence to be relevant. Otherwise, the sentence was deemed irrelevant. This evaluation is not perfect for two reasons: (1) answer key strings do not always appear in relevant sentences.[11], and (2) some arguably relevant sentences may not contain an answer key string (e.g., they may contain a pronoun that refers to the answer, but the pronoun itself is not the desired extraction). However, judging

[11]This happens due to coreference, e.g., when multiple occurrences of an answer appear in a document, some of them may occur in relevant sentences while others do not.

		Irrelevant			Relevant		
	Acc	Rec	Pr	F	Rec	Pr	F
Terrorism							
Iter #1	.84	.93	.89	.91	.41	.55	.47
Iter #2	.84	.90	.91	.90	.54	.51	.53
Iter #3	.82	.85	.92	.89	.63	.46	.53
Disease Outbreaks							
Iter #1	.75	.96	.76	.85	.21	.66	.32
Iter #2	.71	.76	.82	.79	.58	.48	.53
Iter #3	.63	.60	.85	.70	.72	.41	.52

Table 1: Relevant Sentence Classifier Evaluation

the relevance of sentences without relying on answer keys is also tricky, so we decided that this approach was probably good enough to get a reasonable assessment of the classifier. Using this criterion, 17% of the sentences in the terrorism articles are relevant, and 28% of the sentences in the disease outbreaks articles are relevant.

Table 1 shows the accuracy, recall, precision, and F scores of the SVM classifiers after each self-training iteration. The classifiers generated after the third iteration were used in our IE experiments. The final accuracy is 82% in the terrorism domain, and 63% for the disease outbreaks domain. The precision on irrelevant sentences is high in both domains, but the precision on relevant sentences is relatively weak. Despite this, we will show in Section 6.3 that the classifier is effective for the IE task. The reason why the classifier improves IE performance is because it favorably alters the proportion of relevant sentences that are passed along to the IE system. For example, an analysis of the tuning set shows that removing the sentences deemed to be irrelevant by the classifier increases the proportion of relevant sentences from 17% to 46% in the terrorism domain, and from 28% to 41% in the disease outbreaks domain.

We will also see in Section 6.3 that IE recall only drops a little when the sentence classifier is used, despite the fact that its recall on relevant sentences is only 63% in terrorism and 72% for disease outbreaks. One possible explanation is that the answer keys often contain multiple acceptable answer strings (e.g., “John Kennedy” and “JFK” might both be acceptable answers). On average, the answer keys contain approximately 1.64 acceptable strings per answer in the terrorism domain, and 1.77 acceptable strings per answer in the disease outbreaks do-

Patterns	App	Rec	Pr	F	Rec	Pr	F
Terrorism							
		PerpInd			**PerpOrg**		
ASlogTS	All	.49	.35	.41	.33	.49	.40
ASlogTS	Rel	.41	.50	.45	.27	.58	.37
		Target			**Victim**		
ASlogTS	All	.64	.42	.51	.52	.48	.50
ASlogTS	Rel	.57	.49	.53	.48	.54	.51
		Weapon					
ASlogTS	All	.45	.39	.42			
ASlogTS	Rel	.40	.51	.45			
Disease Outbreaks							
		Disease			**Victim**		
ASlogTS	All	.51	.27	.36	.48	.35	.41
ASlogTS	Rel	.46	.31	.37	.44	.38	.41

Table 2: AutoSlog-TS Results

		Disease			**Victim**		
Patterns	*App*	*Rec*	*Pr*	*F*	*Rec*	*Pr*	*F*
ASlogTS	All	.51	.27	.36	.48	.35	.41
SA-50	All	.51	.25	.34	.47	.41	**.44**
SA-50	Rel	.49	.31	**.38**	.44	.43	.43
SA-50	Sel	.50	.29	.36	.46	.41	**.44**
SA-100	All	.57	.22	.32	.52	.33	.40
SA-100	Rel	.55	.28	.37	.49	.36	.41
SA-100	Sel	.56	.26	.35	.51	.34	.41
SA-150	All	.66	.20	.31	.55	.27	.37
SA-150	Rel	.61	.26	.36	.51	.31	.38
SA-150	Sel	.63	.24	.35	.53	.29	.37
SA-200	All	.68	.19	.30	.56	.26	.36
SA-200	Rel	.63	.25	.35	.52	.30	.38
SA-200	Sel	.65	.23	.34	.54	.28	.37

Table 3: ProMed Disease Outbreak Results

main. Thus, even if the sentence classifier discards some relevant sentences, an equally acceptable answer may be found in a different sentence.

6.3 Information Extraction Results

We first conducted two experiments with an existing IE pattern learner, AutoSlog-TS (Riloff, 1996) to give us a baseline against which to compare our results. The "All" rows in Table 2 show these results, where "All" means that the IE patterns were applied to all of the sentences in the test set. AutoSlog-TS[12] produced F scores between 40-51% on the MUC-4 test set, and 36-41% on the ProMed test set. The terrorism scores are competitive with the MUC-4 scores reported by Chieu et al. (2003), although they are not directly comparable because those scores are based on template generation. Since we created the ProMed test set ourselves, we are the first to report results on it[13].

Next, we evaluated the performance of AutoSlog-TS' extraction patterns when they are applied only in the sentences deemed to be relevant by our relevant sentence classifier. The purpose of this experiment was to determine whether the relevant sentence classifier can be beneficial when used with IE patterns known to be of good quality. The "Rel" rows in Table 2 show the scores for this experiment. Precision increased substantially on all 7 roles, although with some recall loss. This shows that a sentence classifier that has a high precision on irrelevant sentences but only a moderate precision on relevant sentences can be useful for information extraction.

Tables 3 and 4 show the results of our IE system, which uses the top N Semantic Affinity (SA) patterns and the relevant sentence classifier. We also show the AutoSlog-TS results again in the top row for comparison. The best F score for each role is shown in boldface. We used a lower probability threshold θ_l of 0.5 to filter out irrelevant patterns. We then ranked the remaining patterns based on semantic affinity, and evaluated the performance of the top 50, 100, 150, and 200 patterns. The *App* column indicates how the patterns were applied: for *All* they were applied in all sentences in the test set, for *Rel* they were applied only in the relevant sentences (as judged by our sentence classifier). For the *Sel* condition, the Primary patterns were applied in all sentences but the Secondary patterns were applied only in relevant sentences. To separate Primary and Secondary patterns we used an upper probability threshold θ_u of 0.8.

Looking at the rows with the *All* condition, we see that the semantic affinity patterns achieve good recall (e.g., the top 200 patterns have a recall over 50% for most roles), but precision is often quite low. This is not surprising because high semantic affinity patterns do not necessarily have to be relevant to the domain, so long as they recognize semantically appropriate things.

[12] AutoSlog-TS was trained on a much larger data set of 4,958 ProMed and 10,191 PubMed documents for the disease outbreaks domain. AutoSlog-TS requires a human review of the top-ranked patterns, which resulted in 396 patterns for the terrorism domain and 125 patterns for the disease outbreaks domain.

[13] Some previous work has been done with ProMed articles (Grishman et al., 2002a; Grishman et al., 2002b), but we are not aware of any IE evaluations on them.

Patterns	App	PerpInd			PerpOrg			Target			Victim			Weapon		
		Rec	*Pr*	*F*	*Rec*	*Pr*	*F*	*Rec*	*Pr*	*F*	*Rec*	*Pr*	*F*	*Rec*	*Pr*	*F*
ASlogTS	All	.49	.35	.41	.33	.49	.40	.64	.42	**.51**	.52	.48	**.50**	.45	.39	.42
SA-50	All	.24	.29	.26	.20	.42	.27	.42	.43	.42	.41	.43	.42	.53	.46	**.50**
SA-50	Rel	.19	.32	.24	.18	.60	.28	.38	.48	.42	.37	.52	.43	.41	.56	.48
SA-50	Sel	.20	.33	.25	.20	.54	.29	.42	.50	.45	.38	.52	.44	.43	.53	.48
SA-100	All	.40	.30	.34	.30	.43	.35	.56	.38	.45	.45	.37	.41	.55	.43	.48
SA-100	Rel	.36	.39	.38	.25	.59	.35	.52	.45	.48	.40	.47	.44	.45	.51	.48
SA-100	Sel	.38	.40	.39	.27	.55	.36	.56	.46	.50	.41	.47	.44	.47	.49	.48
SA-150	All	.50	.27	.35	.34	.39	.37	.62	.30	.40	.50	.33	.40	.55	.39	.45
SA-150	Rel	.46	.39	.42	.28	.58	.38	.56	.37	.45	.44	.45	.45	.45	.50	.47
SA-150	Sel	.48	.39	**.43**	.31	.55	.40	.60	.37	.46	.46	.44	.45	.47	.47	.47
SA-200	All	.73	.08	.15	.42	.43	.42	.64	.29	.40	.54	.32	.40	.64	.17	.27
SA-200	Rel	.67	.15	.24	.34	.61	.43	.58	.36	.45	.47	.43	.45	.52	.29	.37
SA-200	Sel	.71	.12	.21	.36	.58	**.45**	.61	.35	.45	.48	.43	.45	.53	.22	.31

Table 4: MUC-4 Terrorism Results

Next, we can compare each *All* row with the *Rel* row immediately below it. We observe that in every case precision improves, often dramatically. This demonstrates that our sentence classifier is having the desired effect. However, observe that the precision gain comes with some loss in recall points.

Clearly, this drop in recall is due to the answers embedded inside relevant sentences incorrectly classified as irrelevant. To counter this, we apply the Primary patterns to all the sentences. Thus, if we compare each *Rel* row with the *Sel* row immediately below it, we see the effect of loosening the reins on the Primary patterns (the Secondary patterns are still restricted to the relevant sentences). In most cases, the recall improves with a relatively small drop in precision, or no drop at all. In the terrorism domain, the highest F score for four of the five roles occurs under the *Sel* condition. In the disease outbreaks domain, the best F score for diseases occurs in the *Rel* condition, while the best score for victims is achieved under both the *All* and the *Sel* conditions.

Finally, we note that the best F scores produced by our information extraction system are higher than those produced by AutoSlog-TS for all of the roles except Targets and Victims, and our best performance on Targets is only slightly lower. These results are particularly noteworthy because AutoSlog-TS requires a human to manually review the patterns and assign event roles to them. In contrast, our approach is fully automated.

These results validate our hypothesis that decoupling the processes of finding relevant regions and applying semantically appropriate patterns can create an effective IE system.

7 Conclusions

In this work, we described an information extraction system based on a relevant sentence classifier and extraction patterns learned using a *semantic affinity* metric. The sentence classifier was self-trained using only relevant and irrelevant documents plus a handful of seed extraction patterns. We showed that separating the task of relevant region identification from that of pattern extraction can be effective for information extraction. In addition, we observed that the use of a relevant sentence classifier is beneficial for an IE system.

There are several avenues that need to be explored for future work. First, it would be interesting to see if the use of richer features can improve classifier performance, and if that in turn improves the performance of the IE system. We would also like to experiment with different region sizes and study their effect on information extraction. Finally, other techniques for learning semantically appropriate extraction patterns could be investigated.

Acknowledgments

This research was supported by NSF Grant IIS-0208985, Department of Homeland Security Grant N0014-07-1-0152, and the Institute for Scientific Computing Research and the Center for Applied Scientific Computing within Lawrence Livermore National Laboratory. We are grateful to Sean Igo and Rich Warren for annotating the disease outbreaks corpus.

References

E. Agichtein and L. Gravano. 2000. Snowball: Extracting Relations from Large Plain-Text Collections. In *Proceedings of the Fifth ACM Conference on Digital Libraries*, pages 85–94, San Antonio, TX, June.

R. Bunescu and R. Mooney. 2004. Collective Information Extraction with Relational Markov Networks. In *Proceeding of the 42nd Annual Meeting of the Association for Computational Linguistics*, pages 438–445, Barcelona, Spain, July.

M. Califf and R. Mooney. 1999. Relational Learning of Pattern-matching Rules for Information Extraction. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence*, pages 328–334, Orlando, FL, July.

J. Callan. 1994. Passage-Level Evidence in Document Retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 302–310, Dublin, Ireland, July.

H. Chieu, H. Ng, and Y. Lee. 2003. Closing the Gap: Learning-Based Information Extraction Rivaling Knowledge-Engineering Methods. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 216–223, Sapporo, Japan, July.

F. Ciravegna. 2001. Adaptive Information Extraction from Text by Rule Induction and Generalisation. In *Proceedings of Seventeenth International Joint Conference on Artificial Intelligence*, pages 1251–1256, Seattle, WA, August.

C. Fellbaum, editor. 1998. *WordNet: An electronic lexical database*. MIT Press.

D. Freitag and A. McCallum. 2000. Information Extraction with HMM Structures Learned by Stochastic Optimization. In *Proceedings of the Seventeenth National Conference on Artificial Intelligence*, pages 584–589, Austin, TX, August.

D. Freitag. 1998. Toward General-Purpose Learning for Information Extraction. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics*, pages 404–408, Montreal, Quebec, August.

R. Grishman, S. Huttunen, and R. Yangarber. 2002a. Information Extraction for Enhanced Access to Disease Outbreak Reports. *Journal of Biomedical Informatics*, 35(4):236–246, August.

R. Grishman, S. Huttunen, and R. Yangarber. 2002b. Real-Time Event Extraction for Infectious Disease Outbreaks. In *Proceedings of the 3rd Annual Human Language Technology Conference*, San Diego, CA, March.

J. Hobbs, D. Appelt, J. Bear, D. Israel, M. Kameyama, M. Stickel, and M. Tyson. 1997. FASTUS: A Cascaded Finite-state Transducer for Extracting Information for Natural-Language Text. In E. Roche and Y. Schabes, editors, *Finite-State Language Processing*, pages 383–406. MIT Press, Cambridge, MA.

S. Huffman. 1996. Learning Information Extraction Patterns from Examples. In S. Wermter, E. Riloff, and G. Scheler, editors, *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*, pages 246–260. Springer, Berlin.

T. Joachims. 1998. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the Tenth European Conference on Machine Learning*, pages 137–142, April.

J. Kim and D. Moldovan. 1993. PALKA: A System for Lexical Knowledge Acquisition. In *Proceedings of the Second International Conference on Information and Knowledge Management*, pages 124–131, Washington, DC, November.

S. Patwardhan and E. Riloff. 2006. Learning Domain-Specific Information Extraction Patterns from the Web. In *Proceedings of the ACL 2006 Workshop on Information Extraction Beyond the Document*, pages 66–73, Sydney, Australia, July.

A. Popescu, A. Yates, and O. Etzioni. 2004. Class Extraction from the World Wide Web. In Ion Muslea, editor, *Adaptive Text Extraction and Mining: Papers from the 2004 AAAI Workshop*, pages 68–73, San Jose, CA, July.

E. Riloff and W. Phillips. 2004. An Introduction to the Sundance and AutoSlog Systems. Technical Report UUCS-04-015, School of Computing, University of Utah.

E. Riloff. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. In *Proceedings of the Eleventh National Conference on Artificial Intelligence*, pages 811–816, Washington, DC, July.

E. Riloff. 1996. Automatically Generating Extraction Patterns from Untagged Text. In *Proceedings of the Thirteenth National Conference on Articial Intelligence*, pages 1044–1049, Portland, OR, August.

G. Salton, J. Allan, and C. Buckley. 1993. Approaches to Passage Retrieval in Full Text Information Systems. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development on Information Retrieval*, pages 49–58, Pittsburgh, PA, June.

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, Montreal, Canada, August.

S. Soderland. 1999. Learning Information Extraction Rules for Semi-Structured and Free Text. *Machine Learning*, 34(1-3):233–272, February.

K. Sudo, S. Sekine, and R. Grishman. 2003. An Improved Extraction Patterns Representation Model for Automatic IE Pattern Acquisition. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 224–231, Sapporo, Japan, July.

B. Sundheim. 1992. Overview of the Fourth Message Understanding Evaluation and Conference. In *Proceedings of the Fourth Message Understanding Conference (MUC-4)*, pages 3–21, McLean, VA, June.

V. Vapnik. 1995. *The Nature of Statistical Learning Theory*. Springer, New York, NY.

A. Yakushiji, Y. Miyao, T. Ohta, Y. Tateisi, and J. Tsujii. 2006. Construction of Predicate-argument Structure Patterns for Biomedical Information Extraction. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 284–292, Sydney, Australia, July.

R. Yangarber, R. Grishman, P. Tapanainen, and S. Huttunen. 2000. Automatic Acquisition of Domain Knowledge for Information Extraction. In *Proceedings of the 18th International Conference on Computational Linguistics*, pages 940–946, Saarbrücken, Germany, August.

R. Yangarber, W. Lin, and R. Grishman. 2002. Unsupervised Learning of Generalized Names. In *Proceedings of the 19th International Conference on Computational Linguistics*, pages 154–160, Taipei, Taiwan, August.