Using Cases to Represent Context for Text Classification

Ellen Riloff

Department of Computer Science
University of Massachusetts
Amherst, MA 01003
riloff@cs.umass.edu

Abstract

Research on text classification has typically focused on keyword searches and statistical techniques. Keywords alone cannot always distinguish the relevant from the irrelevant texts and some relevant texts do not contain any reliable keywords at all. Our approach to text classification uses case-based reasoning to represent natural language contexts that can be used to classify texts with extremely high precision. The case base of natural language contexts is acquired automatically during sentence analysis using a training corpus of texts and their correct relevancy classifications. A text is represented as a set of cases and we classify a text as relevant if any of its cases are deemed to be relevant. We rely on the statistical properties of the case base to determine whether similar cases are highly correlated with relevance for the domain. Preliminary experiments suggest that case-based text classification can achieve very high levels of precision and outperforms our previous algorithms based on relevancy signatures.

Introduction

Text classification has received considerable attention from the information retrieval community, which has typically approached the problem using keyword searches and statistical techniques [Salton 1989]. However, text classification can be difficult because keywords are often not enough to distinguish the relevant from the irrelevant texts. In previous work [Riloff and Lehnert 1992], we showed that natural language processing techniques can be used to generate *relevancy signatures* which are highly effective at retrieving relevant texts with extremely high precision. Although relevancy signatures capture more information than keywords alone, they still represent only a local context surrounding important words or phrases in a sentence. As a result, they are susceptible to false hits when a correct relevancy discrimination depends upon additional context. Furthermore, many relevant texts cannot be recognized by searching for only key words or phrases. A sentence may contain an abundance of information that clearly makes a text relevant, but does not contain any particular word or phrase that is highly indicative of relevance on its own.

In response to these limitations, we began to explore case-based reasoning as an approach for naturally representing larger pieces of context. Given a training corpus of texts and their correct relevancy classifications, we create a case base that serves as a database of contexts for thousands of sentences from hundreds of texts. Each case represents the natural language context associated with a single sentence. The correct relevancy classifications associated with our training corpus give us the correct classification for each *text*, but not for each *case*. Therefore we cannot merely retrieve the most similar case and apply its classification to the current case. Instead, we retrieve many cases and rely on the statistical properties of the retrieved cases to make an informed decision. Our approach therefore differs from many other CBR systems in that we do not retrieve one or a few very similar cases and apply them directly to the current case (e.g., [Ashley 1990; Hammond 1986]). Instead, we retrieve many cases that are similar to the current case in specific ways and use the statistical properties of the cases as part of a classification algorithm.

In this paper, we show that case-based reasoning is a natural technique for representing and reasoning with natural language contexts for the purpose of text classification. First, we briefly describe the MUC performance evaluations of text analysis systems that stimulated our research on text classification. Second, we introduce *relevancy signatures* and explain how natural language processing techniques can be used effectively for high-precision text classification. Third, we describe our case representation and present the details of the CBR algorithm. We conclude with preliminary test results from an experiment to evaluate the performance of case-based text classification.

Text Classification in MUC-4

Although text classification has typically been addressed by researchers in the information retrieval community, it is an interesting and important problem for natural language processing (NLP) researchers as well. In 1991 and 1992, the UMass natural language processing group participated in the Third and Fourth Message Understanding

Conferences (MUC-3 and MUC-4) sponsored by DARPA. The task for both MUCs was to extract information about terrorism from news wire articles. The information extraction task, however, subsumes the more general problem of text classification. That is, we should know whether a text contains relevant information before we extract anything. We could depend on the information extraction system to do text classification as a side effect: if the system extracts any information at all then we classify the text as relevant. But the information extraction task requires natural language processing capabilities that are costly to apply, even within the constraints of a limited domain. In conjunction with a highly reliable text classification algorithm, however, the text extraction system could be invoked only when a text was classified as relevant, thereby saving the costs associated with applying the full NLP system to each and every text.[1] In addition, information extraction systems generate false hits when they extract information that appears to be relevant to the domain, but is not relevant when the larger context of the text is taken into account. By employing a text classification algorithm at the front end, we could ease the burden on the information extraction system by filtering out many irrelevant texts that might lead it astray.

Motivated by our new found appreciation for text classification, we began working on this task using the MUC-4 corpus as a testbed. The MUC-4 corpus consists of 1700 texts in the domain of Latin American terrorism, as well as a set of templates (answer keys) for each of the texts. These texts were drawn from a database constructed and maintained by the Foreign Broadcast Information Service (FBIS) of the U.S. government and come from a variety of news sources including wire stories, transcripts of speeches, radio broadcasts, terrorist communiqués, and interviews (see [MUC-4 Proceedings 1992]). The templates (answer keys) were generated by the participants of MUC-3 and MUC-4. Each template contains the correct information corresponding to a relevant incident described in a text, i.e. the information that should be extracted for the incident. If a text describes multiple relevant incidents, then the answer keys contain one template for each incident; if a text describes no relevant incidents then the answer keys contain only a dummy template for the text with no information inside. For the purpose of text classification, we use these answer keys to determine the correct relevancy classification of a text. If we find any filled templates among the answer keys for a text then we know that the text contains at least one relevant incident. However, if we find only a dummy template then the text contains no relevant incidents and is irrelevant. Roughly 50% of the texts in the MUC-4 corpus are relevant.

[1]Although our text classification algorithm relies on the sentence analyzer, a complete natural language processing system usually has additional components, such as discourse analysis, that would not need to be applied.

Relevancy Discriminations and Selective Concept Extraction

During the course of MUC-3 and MUC-4, we manually skimmed hundreds of texts in an effort to better understand the information extraction task and to carefully monitor the performance of our system. As we scanned the texts, we noticed that some texts are much easier to classify than others. Although the MUC-4 domain is Latin American terrorism, an extensive set of domain guidelines specifies the details of the domain. For example, events that occurred more than 2 months before the wire date, general event descriptions, and incidents that involve only military targets were not considered to be relevant incidents.

There will always be some texts that are difficult to classify because of exceptions or because they fall into gray areas that are not covered by the domain guidelines. But despite detailed criteria, many relevant texts can be identified quickly and accurately because they contain an event description that clearly falls within the domain guidelines. Often, the presence of a key phrase or sentence is sufficient to confidently classify a text as relevant. For example, the phrase "was assassinated" almost always describes a terrorist event in the MUC-4 corpus. Furthermore, as soon as a relevant incident is identified, the text can be classified as relevant regardless of what appears in the remainder of the text. Therefore, once we hit on a key phrase or sentence, we can completely ignore the rest of the text![2] This property makes the technique of *selective concept extraction* particularly well-suited for the task of relevancy discriminations.

Selective concept extraction is a form of sentence analysis realized by the CIRCUS parser [Lehnert 1990]. The strength behind selective concept extraction is its ability to ignore large portions of text that are not relevant to the domain while selectively bringing in domain knowledge when a key word or phrase is identified. In CIRCUS, selective concept extraction is achieved on the basis of a domain-specific dictionary of *concept nodes* that are used to recognize phrases and expressions that are relevant to the domain of interest. Concept nodes are case frames that are triggered by individual lexical items but activated only in the presence of specific linguistic expressions. For example, in our dictionary the word "dead" triggers two concept nodes: **$found-dead-pass$** and **$left-dead$**. When the word "dead" is found in a text, both concept nodes are triggered. However, **$found-dead-pass$** is activated only if the word preceding "dead" is the verb "found" and the verb appears in a passive construction. Similarly, the concept node **$left-dead$** is activated only if the word "dead" is preceded by the verb

[2]This is not always the case, particularly when the domain description contains exceptions such as the ones mentioned above. Our approach assumes that these exceptional cases are relatively infrequent in the corpus.

"left" and the verb appears in an active construction. Therefore, at most one of these concept nodes will be activated depending upon the surrounding linguistic context. In general, if a sentence contains multiple triggering words then multiple concept nodes may be activated; however, if a sentence contains no triggering words then no output is generated for the sentence. The UMass/MUC-4 dictionary was constructed specifically for MUC-4 and contains 389 concept node definitions (see [Lehnert et al. 1992] for a description of the UMass/MUC-4 system).

Relevancy Signatures

As we noted earlier, many relevant texts can be identified by recognizing a single but highly relevant phrase or expression, e.g. "X was assassinated". However, keywords are often not enough to reliably identify relevant texts. For example, the word "dead" is a very strong relevancy cue for the terrorism domain in some contexts but not in others. Phrases of the form "<person> was found dead" are highly associated with terrorism in the MUC-4 corpus because "found dead" has an implicit connotation of foul play; in fact, every text in the MUC-4 corpus that contains an expression of this form is a relevant text. However expressions of the form "left <number> dead", as in "the attack left 49 dead", are often used in military event descriptions that are not terrorist in nature. Therefore these phrases are not highly correlated with relevant texts in the MUC-4 corpus. As a consequence, the keyword "dead" is not a good indicator of relevance on its own, even though certain expressions containing the word "dead" *are* very strong relevancy cues that can be used reliably to identify relevant texts.

In previous work, we introduced the notion of a *relevancy signature* to recognize key phrases and expressions that are strongly associated with relevance for a domain. A *signature* is a pair consisting of a lexical item and a concept node that it triggers, e.g. <placed, **$loc-val-pass-1$**>. Together, this pair recognizes the set of expressions of the form: "<weapon> <auxiliary> placed", such as, "a bomb was placed" or "dynamite sticks were placed". The word "placed" triggers the concept node **$loc-val-pass-1$** which is activated only if it appears in a passive construction and the subject of the clause is a weapon. Note that different words can trigger the same concept node and the same word can trigger multiple concept nodes. For example, the signature <planted, **$loc-val-pass-1$**> recognizes passive constructions such as "a bomb was planted" and <planted, **$loc-val-1$**> recognizes active constructions such as "the terrorists planted a bomb". Finally, *relevancy signatures* are signatures that are highly correlated with relevant texts in a training corpus. The presence of a single relevancy signature in a text is often enough to classify the text as relevant.

In a set of experiments with the MUC-3 corpus [Riloff and Lehnert 1992], we showed that relevancy signatures can be highly effective at identifying relevant texts with high precision. However, we noticed that many of their errors were due to relevancy classifications that depended upon additional context surrounding the key phrases. For example, two very similar sentences (1) "a civilian was killed by guerrillas" and (2) "a soldier was killed by guerrillas" are represented by the same signature <killed, **$kill-pass-1$**>. However, (1) describes a relevant incident and (2) does not because our domain definition specifies that incidents involving military targets are not acts of terrorism. Relevancy signatures alone cannot make distinctions that depend on the slot fillers inside the concept nodes, such as the victims and perpetrators.

With this in mind, we extended our work on relevancy signatures by augmenting them with slot filler information from inside the concept nodes. Whereas relevancy signatures represent the existence of concept nodes, *augmented relevancy signatures* represent entire concept node instantiations. The augmented relevancy signatures algorithm [Riloff and Lehnert 1992] classifies texts by looking for both a strong relevancy signature *and* a reliable slot filler inside the concept node. For example, "a civilian was killed" activates a murder concept node that extracts the civilian as its victim. The augmented relevancy signatures algorithm will classify this as a highly relevant expression only if its signature <killed, **$kill-pass-1$**> is highly correlated with relevance and its slot filler (the civilian victim) is highly correlated with relevance as well.

Additional experiments with augmented relevancy signatures demonstrated that they can achieve higher precision text classification than relevancy signatures alone [Riloff and Lehnert 1992]. However, augmented relevancy signatures can still fall short when relevancy discriminations depend on multiple slot fillers in a concept node or on information that is scattered throughout a sentence. Furthermore, some relevant sentences do not contain *any* key phrases that are highly associated with relevance for a domain. Instead, a sentence may contain many pieces of information, none of which is individually compelling, but which in total clearly describe a relevant incident. For example, consider this sentence:

Police sources have confirmed that a guerrilla was killed and two civilians were wounded this morning during an attack by urban guerrillas.

This sentence clearly describes a terrorist incident even though it does not contain any key words or phrases that are highly indicative of terrorism individually. The important action words, "killed", "wounded", and "attack", all refer to a violent event but not necessarily a terrorist event; e.g., people are often killed, wounded, and attacked in military incidents (which are prevalent in the MUC-4 corpus). Similarly, "guerrillas" is a reference to terrorists but "guerrillas" are mentioned in many texts that do not describe a specific terrorist act. Collectively, however, the entire context clearly depicts a terrorist event because it contains relevant action words, relevant victims (civilians), and relevant perpetrators (guerrillas). We need

all of this information to conclude that the incident is relevant. These observations prompted us to pursue a different approach to text classification using case-based reasoning. By representing the contexts of entire sentences with cases, we can begin to address many of the issues that were out of our reach using only relevancy signatures.

Case-based Text Classification

The motivation behind our case-based approach is twofold: (1) to more accurately classify texts and (2) to classify texts that are inaccessible via techniques that focus only on key words or phrases. We will present the CBR algorithm in several steps. First, we describe our case representation and explain how we generate cases from texts. Second, we introduce the concept of a *relevancy index* and show how these indices allow us to retrieve cases that might have been inaccessible via key words or phrases alone. And finally, we present the details of the algorithm itself.

The Case Representation

We represent a text as a set of cases using the following procedure. Given a text to classify, we analyze each sentence using CIRCUS and collect the concept nodes that are produced during sentence analysis. Then we create a case for each sentence by merging all of the concept nodes produced by the sentence into a single structure. A case is represented as a frame with five slots: *signatures*, *perpetrators*, *victims*, *targets*, and *instruments*. The *signatures* slot contains the signatures associated with each concept node in the sentence. The remaining four slots represent the slot fillers that were picked up by these concept nodes.[3] Although the concept nodes extract specific strings from the text, e.g. "the guerrillas", a case retains only the semantic features of the fillers, e.g. `terrorist`, in order to generalize over the specific strings that appeared in the text. A sample sentence, the concept nodes produced by the sentence, and its corresponding case are shown below:

Two vehicles were destroyed and an unidentified office of the agriculture and livestock ministry was heavily damaged following the explosion of two bombs yesterday afternoon.

This sentence generates three concept nodes:

(1) **$DESTRUCTION-PASS-1$** (triggered by *destroyed*)
TARGET = *two vehicles*

(2) **$DAMAGE-PASS-1$** (triggered by *damaged*)
TARGET = *an unidentified office of the agriculture and livestock ministry*

(3) **$WEAPON-BOMB$** (triggered by *bombs*)

These concept nodes result in the following case:

CASE
SIGNATURES = (<destroyed, **$destruction-pass-1$**>
<damaged, **$damage-pass-1$**>
<bombs, **$weapon-bomb$**>)
PERPETRATORS = nil
VICTIMS = nil
TARGETS = (`govt-office-or-residence transport-vehicle`)
INSTRUMENTS = (`bomb`)

Note that our case representation does not preserve the associations between the concept nodes and their fillers, e.g. the case doesn't tell us whether the `govt-office-or-residence` was damaged and the `transport-vehicle` destroyed or vice versa. We purposely disassociated the fillers from their concept nodes so that we can look for associations between concept nodes and their own fillers as well as fillers from other concept nodes. For example, every case in our case base that contains the signature <killing, **$murder-1$**> and a `govt-office-or-residence` target came from a relevant text. The word "killing" does not necessarily describe a terrorist event, but if the incident also involves a `govt-office-or-residence` target[4] then the target seems to provide additional evidence that suggests terrorism. But only human targets can be murdered, so the concept node **$murder-1$** will never contain any physical targets. Therefore, the `govt-office-or-residence` must have been extracted by a *different* concept node from elsewhere in the sentence. The ability to pair slot fillers with *any* signature allows us to recognize associations that may exist between pieces of information from different parts of the sentence.

Relevancy Indices

Since a text is represented as a set of cases, our goal is to determine if any of its cases are relevant to the domain. We will deem a *case* to be relevant if it is similar to other cases in the case base that were found in relevant texts. However, an individual case contains many signatures and slot fillers that could be used to index the case base. It is unlikely that we will find many exact matches by indexing on all of these features, so we use the notion of a relevancy

[3] We chose these four slots because the concept nodes in our UMass/MUC-4 system [Lehnert et al. 1992] use these four types of slots to extract information about terrorism. In general, a case should have one slot for each possible type of concept node slot.

[4] Government officials and facilities are frequently the targets of terrorist attacks in the MUC-4 corpus.

index to retrieve cases that share particular properties with the current case. A *relevancy index* is a triple of the form: <signature, slot filler pair, case outline>. A slot filler pair consists of a slot name and a semantic feature associated with legitimate slot fillers for the slot, e.g. (perpetrators `terrorist`). A case outline is a list of slots in the case that have fillers. For example, the case outline (perpetrators victims) means that these two slots are filled but the remaining slots, targets and instruments, are not. The signature slot is always filled so we do not include it as part of the case outline.

This index represents much of the context of the sentence. As we explained earlier, a signature represents a set of phrases that activate a specific concept node. By indexing on both a signature and a slot filler, we are retrieving cases that represent similar phrases *and* similar slot fillers. Intuitively, this means that the retrieved cases share both a key phrase and a key piece of information that was extracted from the text. The ability to index on both signatures and slot fillers simultaneously allows us to recognize relevant cases that might be missed by indexing on only one of them. For example, the word "assassination" is highly indicative of relevance in our domain because assassinations in Latin America are almost always terrorist in nature. However, the word "assassination" by itself is not necessarily a good keyword because many texts contain generic references to an assassination but do not describe a specific event. Therefore we also need to know that the text mentioned a specific victim, although almost any reference to a victim is good enough. Even a vague description of a victim such as "a person" is enough to confidently classify the text as relevant because the word "assassination" carries so much weight. On the other hand, a phrase such as "the *killing* of a person" is not enough to classify a text as relevant because the word "killing" is not a strong enough relevancy cue for terrorism to offset the vagueness of the victim description.

Similarly, some slot fillers are highly indicative of relevance in the MUC-4 corpus regardless of the event type. For example, when a government official is the victim of a crime in Latin America, it is almost always the result of a terrorist action. Of course, every mention of a government official in a text does not signal relevance although almost any reference to a government official as the victim of a crime is generally good enough. Therefore a phrase such as "a government official was killed" is enough to confidently classify a text as relevant because government officials are so often the victims of terrorism. Alternatively, a phrase such as "a *person* was killed" is not enough to classify a text as relevant since the person could have been killed in many ways that had nothing to do with terrorism. The augmented relevancy signatures algorithm also classifies texts on the basis of both signatures and slot fillers, but both the signature and slot filler must be highly correlated with relevance independently. As a result, augmented relevancy signatures cannot recognize relationships in which a signature and slot filler together represent a relevant situation even though one or both of them are not highly associated with relevance by themselves.

Finally, the third part of a relevancy index is the case outline. A case outline represents the set of slots that are filled in a case. The purpose of the case outline is to allow different signatures to require varying amounts of information. For example, some words such as "assassination" are so strongly correlated with relevance that we don't need much additional information to confidently assume that it refers to a terrorist event. As we described above, the presence of almost any victim is enough to imply relevance. In particular, we don't need to verify that the perpetrator was a terrorist -- the connotations associated with "assassination" are strong enough to make that assumption safely. However, other words, e.g. "killed", do not necessarily refer to a terrorist act and are often used in generic event descriptions such as "many people have been killed in the last year". The presence of a perpetrator, therefore, can signal the difference between a specific incident and a general event description. To illustrate the power of the case outline, consider the following statistics drawn from a case base constructed from 1500 texts. We retrieved all cases from the case base that contained the following relevancy indices and calculated the percentage of the retrieved cases that came from relevant texts:

(<assassination, **$murder-1$**>, (victims `civilian`), (victims))	100%
(<killed, **$kill-pass-1$**>, (victims `civilian`), (victims))	68%
(<killed, **$kill-pass-1$**>, (victims `civilian`), (victims perpetrators))	100%

These statistics clearly show that the "assassination" of civilian victims(s) is strongly correlated with relevance (100%) even without any reference to a perpetrator. But texts that contain civilian victim(s) who are "killed" are not highly correlated with relevance (68%) since the word "killed" is much less suggestive of terrorism. However, texts that contain civilian victim(s) who are "killed" and name a specific perpetrator are highly correlated with relevance (100%). To reliably depend on the word "killed", the presence of a known perpetrator is critical.

By combining a signature, slot filler pair, and case outline into a single index, we retrieve cases that share similar key phrases, at least one similar piece of extracted information, and contain roughly the same amount of information. However, most cases contain many signatures and many slot fillers so we are still ignoring many potentially important differences between the cases. The CBR algorithm described below shows how we try to ensure that these differences are not likely to be important by (1) relying on statistics to assure us that we've hit on a highly reliable index and (2) employing secondary

signature and slot filler checks to ferret out any obvious irrelevancy cues that might turn an otherwise relevant case into an irrelevant one.

The Case-Based Text Classification Algorithm

A text is represented as a set of cases, one case for each sentence that produced at least one concept node. To classify a text, we classify each of its cases in turn until we find a case that is deemed to be relevant or until we exhaust all of its cases. As soon as we identify a relevant case, we immediately classify the entire text as relevant. The strength of the algorithm, therefore, rests on its ability to identify a relevant case. We classify a case as relevant if and only if the following three conditions are satisfied:

Condition 1: The case contains a strong *relevancy index.*
Condition 2: The case does not have any "bad" signatures.
Condition 3: The case does not have any "bad" slot fillers.

Condition 1 is the heart of the algorithm; conditions 2 and 3 are merely secondary checks to recognize irrelevancy cues, i.e. signatures or slot fillers that might turn an otherwise relevant case into an irrelevant one. We'll give some examples of these situations later. First, we'll explain how we use the relevancy indices. Given a target case to classify, we generate each possible relevancy index for the case and, for each index, retrieve all cases from the case base that share the same index. Then we calculate two statistics over the set of retrieved cases: [1] the total number of retrieved cases (N) and [2] the number of retrieved cases that are relevant (N_R). The ratio of N_R/N gives us a measure of the association between the retrieved cases and relevance, e.g. .85 means that 85% of the retrieved cases are relevant. If this relevancy ratio is high then we assume that the index is a strong indicator of relevance and therefore the target case is likely to be relevant as well. More specifically, we use two thresholds to determine whether the retrieved cases satisfy our relevancy criteria: a relevancy threshold (R) and a minimum number of occurrences threshold (M). If the ratio (N_R/N) >= R and N >= M then the cases, and therefore the relevancy index, satisfy Condition 1.

The second threshold, M, is necessary because we must retrieve a reasonably large number of cases to feel confident that our statistics are meaningful. Each case in the case base is tagged with a relevancy classification denoting whether it was found in a relevant or irrelevant text. However, this classification does not necessarily apply to the case itself. If the case is tagged as irrelevant (i.e., it was found in an irrelevant text), then the case itself *must* be irrelevant. However, if the case is tagged as relevant (i.e., it was found in a relevant text) then we can't be sure whether the case is relevant or not. It might very well be a relevant case. On the other hand, it could be an irrelevant case that happened to be in a relevant text because another case in the text was relevant. This is a classic example of the credit assignment problem where we know the correct classification of the text but do not know which case(s) were responsible for that classification.

Because of the fuzziness associated with the case classifications, we cannot rely on a single case or even a small set of cases to give us definitive guidance. Therefore we cannot merely retrieve the most similar case, or a small set of extremely similar cases, as do many other CBR systems. Instead, we retrieve a large number of cases that share certain features with the target case and rely on the statistical properties of these cases to tell us whether these features are correlated with relevance. If a high percentage of the retrieved cases are classified as relevant, then we assume that these common features are responsible for their relevance and we should therefore classify the target case as relevant as well.

Finally, two additional conditions must be satisfied by the target case before we classify it as relevant. Conditions 2 and 3 make sure that the case does not contain any "bad" signatures or slot fillers that might indicate that the case is irrelevant despite other apparently relevant information. For example, the MUC-4 domain guidelines dictate that specific terrorist incidents are relevant but general descriptions of incidents are not. To illustrate the difference, the following sentence is irrelevant to our domain because it contains a summary description of many events but does not describe any single incident:

More than 100 people have died in Peru since 1980, when the Maoist Shining Path organization began its attacks and its wave of political violence.

CASE
SIGNATURES = (<died, **die**>
<wave, **$generic-event-marker$**>
<attacks, **$attack-noun-1$**>)
PERPETRATORS = nil
VICTIMS = (human)
TARGETS = nil
INSTRUMENTS = nil

However a similar sentence is relevant:

More than 100 people have died in Peru during 2 attacks yesterday.

CASE
SIGNATURES = (<died, **die**>
<attacks, **$attack-noun-1$**>)
PERPETRATORS = nil
VICTIMS = (human)
TARGETS = nil
INSTRUMENTS = nil

These sentences generate almost identical cases except that the first sentence contains a signature for a **$generic-event-marker$** triggered by the word "wave" and the

second one does not. Our system relies on a small set of special concept nodes like this one to recognize textual cues that signal a general event description. The presence of this single concept node indicates that the case is irrelevant, even though the rest of the case contains perfectly good information that would otherwise be relevant. Condition 2 of the algorithm recognizes signatures for concept nodes, such as this one, that are only weakly correlated with relevance and therefore might signal that the case is irrelevant.

Similarly, a single slot filler can turn an otherwise relevant case into an irrelevant one. For example, according to the MUC-4 domain guidelines if the perpetrator of a crime is a civilian then the incident is not terrorist in nature. Therefore an incident with a terrorist perpetrator is relevant even though a similar incident with a civilian perpetrator is irrelevant. Condition 3 of the algorithm recognizes slot fillers that are weakly correlated with relevance and therefore might indicate that an event is irrelevant.

We use two additional thresholds, I_{sig} and I_{slot}, to identify "bad" signatures and slot fillers. To evaluate condition 2, for each signature in the target case, we retrieve all cases that also contain that signature and compute N_R and N. If the ratio $(N_R/N) >= I_{sig}$ and $N >= M$ then the cases satisfy Condition 2. To evaluate Condition 3, for each slot filler in the case, we retrieve all cases that also contain that slot filler and the same case outline. Similarly, we compute N_R and N for the retrieved cases and if the ratio $(N_R/N) >= I_{slot}$ and $N >= M$ then the cases satisfy Condition 3.

Conclusions

We have conducted preliminary experiments to compare the performance of case-based text classification with relevancy signatures. Using a case base containing 7032 cases derived from 1500 training texts and their associated answer keys from the MUC-4 corpus, we applied the algorithm to two blind test sets of 100 texts each. These are the same test sets, DEV-0401 and DEV-0801, that we used to evaluate the performance of relevancy signatures [Riloff and Lehnert 1992]. The most notable improvement was that our case-based algorithm achieved much higher precision for DEV-0801: for some parameter settings, case-based text classification correctly identified 41% of the relevant texts with 100% precision, whereas augmented relevancy signatures could correctly identify only 8% of the relevant texts with 100% precision and relevancy signatures alone could not obtain 100% precision on this test set at all. Performance on the other test set, DEV-0401, was comparable for both augmented relevancy signatures and the case-based algorithm, both of which slightly outperformed relevancy signatures alone. We suspect that we are seeing a ceiling effect on this test set since all of the algorithms do extremely well with it.

Our work differs from many CBR systems in that we do not retrieve a single best case or a few highly similar cases and apply them directly to the new case (e.g., [Ashley 1990; Hammond 1986]). Instead, we retrieve a large number of cases that share specific features with the new case and use the statistical properties of the retrieved cases as part of a classification algorithm. MBRtalk [Stanfill and Waltz 1986] and PRO [Lehnert 1987] are examples of other systems that have worked with large case bases. MBRtalk is similar to many CBR systems in that it applies a similarity metric to retrieve a few best matches (10) and uses these cases to determine a response. PRO, on the other hand, is closer in spirit to our approach because it relies on frequency data from the case base to drive the algorithm. PRO builds a network from the entire case base and applies a relaxation algorithm to the network to generate a response. The activation levels of certain nodes in the network are based upon frequency data from the case base. Although both PRO and our system rely on statistics from the case base, our approach differs in that we depend on the statistics, not only to find the best response, but also to resolve the credit assignment problem because the classifications of our cases are not always certain.

Finally, our case-based text classification algorithm is domain-independent so the system can be easily scaled up and ported to new domains by simply extending or replacing the case base. To generate a case base for a new domain, CIRCUS needs a domain-specific dictionary of concept nodes. Given a training corpus for the domain, we have shown elsewhere that the process of creating a domain-specific dictionary of concept nodes for text extraction can be successfully automated [Riloff 1993; Riloff and Lehnert 1993].[5] However, our approach does assume that the lexicon contains semantic features, at least for the words that can be legitimate slot fillers for the concept nodes. Given only a set of training texts, their correct relevancy classifications, and a domain-specific dictionary, the case base is acquired automatically as a painless side effect of sentence analysis.

We should also point out that this approach does not depend on an explicit set of domain guidelines. The case base implicitly captures domain specifications which are automatically derived from the training corpus. This strategy simplifies the job of the knowledge engineer considerably. The domain expert only needs to generate a training corpus of relevant and irrelevant texts, as opposed to a complex set of domain guidelines and specifications.

As storage capacities grow and memory becomes cheaper, we believe that text classification will become a central problem for an increasing number of computer applications and users. And as more and more documents become accessible on-line, we expect that high-precision text classification will become paramount. Using relevancy indices, we retrieve texts that share similar key

[5] Although the special concept nodes used to identify "bad" signatures for condition 2 are an exception. The concept nodes that we acquire automatically are not designed to recognize expressions that are negatively correlated with the domain.

phrases, at least one similar piece of extracted information, and contain roughly the same amount of information. We are encouraged by our results so far which demonstrate that these rich natural language contexts can be used to achieve extremely accurate text classification.

Acknowledgments

The author would like to thank Claire Cardie and Wendy Lehnert for providing helpful comments on earlier drafts of this paper. This research was supported by the Office of Naval Research Contract N00014-92-J-1427 and NSF Grant no. EEC-9209623, State/Industry/University Cooperative Research on Intelligent Information Retrieval.

Bibliography

Ashley, K. 1990. Modelling Legal Argument: Reasoning with Cases and Hypotheticals. Cambridge, MA. The MIT Press.

Hammond, K. 1986. CHEF: A Model of Case-Based Planning. In *Proceedings of the Fifth National Conference on Artificial Intelligence*, pp. 267-271. Morgan Kaufmann.

Lehnert, W. G. 1987. Case-Based Problem Solving with a Large Knowledge Base of Learned Cases. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, pp. 301-306. Morgan Kaufmann.

Lehnert, W. G. 1990. Symbolic/Subsymbolic Sentence Analysis: Exploiting the Best of Two Worlds. In *Advances in Connectionist and Neural Computation Theory*. (Eds: J. Pollack and J. Barnden), pp. 135-164. Norwood, NJ. Ablex Publishing.

Lehnert, W., Cardie, C., Fisher, D., McCarthy, J., Riloff, E., and Soderland, S. 1992. University of Massachusetts: Description of the CIRCUS System as Used for MUC-4. In *Proceedings of the Fourth Message Understanding Conference*, pp. 282-288.

Proceedings of the Fourth Message Understanding Conference. 1992. San Mateo, CA. Morgan Kaufmann.

Riloff, E. 1993. Automatically Constructing a Dictionary for Information Extraction Tasks. Submitted to the Eleventh National Conference on Artificial Intelligence.

Riloff, E. and Lehnert, W. 1992. Classifying Texts Using Relevancy Signatures. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, pp. 329-334. AAAI Press/The MIT Press.

Riloff, E. and Lehnert, W. 1993. Automated Dictionary Construction for Information Extraction from Text. To appear in Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications.

Salton, G. 1989. *Automatic Text Processing: The Transformation, Analysis, and Retrieval of Information by Computer*. Reading, MA. Addison-Wesley.

Stanfill, C. and Waltz, D. 1986. Toward Memory-Based Reasoning. *Communications of the ACM*, Vol. 29, No. 12, pp. 1213-1228.