In Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 130-136.

Little Words Can Make a Big Difference for Text Classification

Ellen Riloff
Department of Computer Science
University of Utah
Salt Lake City, UT 84112
E-mail: riloff@cs.utah.edu

Abstract

Most information retrieval systems use stopword lists and stemming algorithms. However, we have found that recognizing singular and plural nouns, verb forms, negation, and prepositions can produce dramatically different text classification results. We present results from text classification experiments that compare *relevancy signatures*, which use local linguistic context, with corresponding indexing terms that do not. In two different domains, relevancy signatures produced better results than the simple indexing terms. These experiments suggest that stopword lists and stemming algorithms may remove or conflate many words that could be used to create more effective indexing terms.

Introduction

Most information retrieval systems use a stopword list to prevent common words from being used as indexing terms. Highly frequent words, such as determiners and prepositions, are not considered to be content words because they appear in virtually every document. Stopword lists are almost universally accepted as a necessary part of an information retrieval system. For example, consider the following quote from a recent information retrieval textbook:

> "It has been recognized since the earliest days of information retrieval (Luhn 1957) that many of the most frequently occurring words in English (like "the","of", "and","to",etc.) are worthless indexing terms." ([Frakes and Baeza-Yates, 1992], p. 113)

Many information retrieval systems also use a stemming algorithm to conflate morphologically related words into a single indexing term. The motivation behind stemming algorithms is to improve recall by generalizing over morphological variants. Stemming algorithms are commonly used, although experiments to determine their effectiveness have produced mixed results (e.g., see [Harman, 1991; Krovetz, 1993]).

One benefit of stopword lists and stemming algorithms is that they significantly reduce the storage requirements of inverted files. But at what price? We have found that some types of words, which would be removed by stopword lists or merged by stemming algorithms, play an important role in making certain domain discriminations. For example, similar expressions containing different prepositions and auxiliary verbs behave very differently. We have also found that singular and plural nouns produce dramatically different text classification results.

First, we will describe a text classification algorithm that uses linguistic expressions called "relevancy signatures" to classify texts. Next, we will present results from text classification experiments in two domains which show that similar signatures produce substantially different classification results. Finally, we discuss the implications of these results for information retrieval systems.

Relevancy Signatures

Relevancy signatures represent linguistic expressions that can be used to classify texts for a specific domain (i.e., topic). The linguistic expressions are extracted from texts automatically using an information extraction system called CIRCUS. The next section gives a brief introduction to information extraction and the CIRCUS sentence analyzer, and the following section describes relevancy signatures and how they are used to classify texts.

Information Extraction

CIRCUS [Lehnert, 1991] is a conceptual sentence analyzer that extracts domain-specific information from text. For example, in the domain of terrorism, CIRCUS can extract the names of perpetrators, victims, targets, weapons, dates, and locations associated with terrorist incidents. Information is

extracted using a dictionary of domain-specific structures called *concept nodes*. Each concept node recognizes a specific linguistic pattern and uses the pattern as a template for extracting information.

For example, a concept node dictionary for the terrorism domain contains a concept node called $murder-passive-victim$ which is triggered by the pattern "<X> was murdered" and extracts X as a murder victim. A similar concept node called $murder-active-perpetrator$ is triggered by the pattern "<X> murdered ..." and extracts X as the perpetrator of a murder. A concept node is activated during sentence processing when it recognizes its pattern in a text.

Figure 1 shows a sample sentence and instantiated concept nodes produced by CIRCUS. Two concept nodes are generated in response to the passive form of the verb "murdered". One concept node, $murder-passive-victim$, extracts the "three peasants" as murder victims, and a second concept node, $murder-passive-perpetrator$, extracts the "guerrillas" as perpetrators. [1]

Sentence:
Three peasants were murdered by guerrillas.

$murder-passive-victim$
victim = "three peasants"

$murder-passive-perpetrator$
perpetrator = "guerrillas"

Figure 1: Two instantiated concept nodes

Theoretically, concept nodes can be arbitrarily complex but, in practice, most of them recognize simple linguistic constructs. Most concept nodes represent one of the general linguistic patterns shown in Figure 2.[2]

All of the information extraction done by CIRCUS happens through concept nodes, so it is crucial to have a good concept node dictionary for a domain. Multiple concept nodes may be generated for a sentence, or no concept nodes may be generated at all. Sentences that do not activate any concept nodes are effectively ignored.

Building a concept node dictionary by hand can be extremely time-consuming and tedious. We estimate that it took approximately 1500 person-hours for two experienced system developers to build a concept node dictionary by hand for the terrorism domain. However, we have since developed a system called AutoSlog [Riloff, 1994; Riloff, 1993] that creates concept node dictionaries automatically using an annotated training corpus. Given a training corpus for the terrorism domain, a dictionary created by AutoSlog achieved 98% of the performance of the hand-crafted dictionary and required only 5 person-hours to build.

Linguistic Pattern	Example
<**subject**> **passive-verb**	<entity> was formed
<**subject**> **active-verb**	<entity> linked
<**subject**> **verb dobj**	<entity> completed acquisition
<**subject**> **verb infinitive**	<entity> agreed to form
<**subject**> **auxiliary noun**	<entity> is conglomerate
active-verb <**dobj**>	acquire <entity>
infinitive <**dobj**>	to acquire <entity>
verb infinitive <**dobj**>	agreed to establish <entity>
gerund <**dobj**>	producing <product>
noun auxiliary <**dobj**>	partner is <entity>
noun prep <**np**>	partnership between <entity>
active-verb prep <**np**>	buy into <entity>
passive-verb prep <**np**>	was signed between <entity>
infinitive prep <**np**>	to collaborate on <product>

Figure 2: Concept node patterns and examples from the joint ventures domain

Relevancy Signatures

Motivation Most information retrieval systems classify texts on the basis of multiple words and phrases. However, for some classification tasks, classifying texts on the basis of a single linguistic expression can be effective. Although single words do not usually provide enough context to be reliable indicators for a domain, slightly larger phrases can be reliable. For example, the word "dead" is not a reliable keyword for murder because people die in many ways that have nothing to do with murder. However, some expressions containing the word "dead" *are* reliable indicators of murder. Figure 3 shows several expressions involving the words "dead" and "fire", and the percentage of occurrences of each expression that appeared in relevant texts. These results are based on 1500 texts from the MUC-4 corpus.[3] The texts in the MUC-4 corpus were retrieved from a general database because they contain one or more words related to terrorism but only half of them actually describe a relevant terrorist incident.[4]

Figure 3 shows that every occurrence of the expression "was found dead" appeared in a relevant text. However, only

[1] In principle, a single concept node can extract more than one item. However, concept nodes produced by AutoSlog [Riloff, 1994; Riloff, 1993] extract only one item at a time. The joint venture results presented in this paper are based on a concept node dictionary produced by AutoSlog.

[2] These are the linguistic patterns used by AutoSlog to create the joint ventures dictionary (see [Riloff, 1994; Riloff, 1993] for details). The concept node dictionary for the terrorism domain was hand-crafted and contains some more complicated patterns as well.

[3] MUC-4 was the Fourth Message Understanding Conference held in 1992 [MUC-4 Proceedings, 1992].

[4] The MUC-4 organizers defined terrorism according to a complicated set of guidelines but, in general, a relevant event was a specific incident that occurred in Latin America involving a terrorist perpetrator and civilian target.

Expression	Rel. %
was found dead	100%
left dead	61%
<number> dead	47%
set on fire	100%
opened fire	87%
<weapon> fire	59%

Figure 3: Strength of Associations for Related Expressions

61% of the occurrences of the expression "left dead" and 47% of the occurrences of "<number> dead" (e.g., "there were 61 dead") appeared in relevant texts. This is because the expression "was found dead" has an implicit connotation of foul play, which suggests that murder is suspected. In contrast, the expressions "left dead" and "<number> dead" often refer to military casualties that are not terrorist in nature.

Figure 3 also shows that several expressions involving the word "fire" have different correlations with relevance. The expression "set on fire" was strongly correlated with relevant texts describing arson incidents, and the expression "opened fire" was highly correlated with relevant texts describing terrorist shooting incidents. However, the expression "<weapon> fire" (e.g., "rifle fire" or "gun fire") was not highly correlated with terrorist texts because it often appears in texts describing military incidents.

These results show that similar linguistic expressions can have very different associations with relevance for a domain. Furthermore, many of these distinctions would be difficult, if not impossible, for a human to anticipate. Based on these observations, we developed a text classification algorithm that automatically identifies linguistic expressions that are strongly associated with a domain and uses them to classify new texts. Our approach uses an underlying information extraction system, CIRCUS, to recognize linguistic context.

The Relevancy Signatures Algorithm A *signature* is defined as a pair consisting of a word and a concept node triggered by that word. Each signature represents a unique set of linguistic expressions. For example, the signature <murdered, $murder-passive-victim$> represents all expressions of the form "was murdered", "were murdered", "have been murdered", etc. Signatures are generated automatically by applying CIRCUS to a text corpus.

A relevancy signature is a signature that is highly correlated with relevant texts in a preclassified training corpus. To generate relevancy signatures for a domain, the training corpus is processed by CIRCUS, which produces a set of instantiated concept nodes for each text. Each concept node is then transformed into a signature by pairing the name of the concept node with the word that triggered it. Once a set of signatures has been acquired from the corpus, for each signature we estimate the conditional probability that a text is relevant given that it contains the signature. The formula is:

$$\Pr(\frac{text\ is\ relevant}{text\ contains\ sig_i}) = \frac{N_{sig_i \epsilon REL-TEXTS}}{N_{sig_i}}$$

where N_{sig_i} is the number of occurrences of the signature sig_i in the training corpus and $N_{sig_i \epsilon REL-TEXTS}$ is the number of occurrences of the signature sig_i in relevant texts in the training corpus. The epsilon is used loosely to denote the number of occurrences of the signature that "appeared in" relevant texts.

Finally, two thresholds are used to identify the signatures that are most highly correlated with relevant texts. A relevance threshold R selects signatures with conditional probability $\geq$ R, and a frequency threshold M selects signatures that have appeared at least M times in the training corpus. For example, R = .85 specifies that at least 85% of the occurrences of a signature in the training corpus appeared in relevant texts, and M = 3 specifies that the signature must have appeared at least 3 times in the training corpus.

To classify a new text, the text is analyzed by CIRCUS and the resulting concept nodes are transformed into signatures. Then the signatures are compared with the list of relevancy signatures for the domain. If any of the relevancy signatures are found, then the text contains an expression that is strongly associated with the domain so it is classified as relevant. If no relevancy signatures are found, then the text is classified as irrelevant. The presence of a single relevancy signature is enough to produce a relevant classification.

Experimental Results for Similar Expressions

Previous experiments demonstrated that the relevancy signatures algorithm can achieve high-precision text classification and performed better than an analogous word-based algorithm in two domains: terrorism and joint ventures (see [Riloff, 1994; Riloff and Lehnert, 1994] for details). In this paper, we focus on the effectiveness of similar linguistic expressions for classification. In many cases, similar signatures generated substantially different conditional probabilities. In particular, we found that recognizing singular and plural nouns, different verb forms, negation, and prepositions was critically important in both the terrorism and joint ventures domains. The results are based on 1500 texts from the MUC-4 terrorism corpus and 1080 texts from a joint ventures corpus.[5] In both corpora, roughly 50% of the texts were relevant to the targeted domain. Although most general-purpose corpora contain a much smaller percentage of relevant texts, our goal is to simulate a pipelined system in which a traditional information retrieval system is first applied to a general-purpose corpus to identify potentially relevant texts. This prefiltered corpus is then used by our system

[5] These texts were randomly selected from a corpus of 1200 texts, of which 719 came from the MUC-5 joint ventures corpus [MUC-5 Proceedings, 1993] and 481 came from the Tipster detection corpus [Tipster Proceedings, 1993; Harman, 1992] (see [Riloff, 1994] for details of how these texts were chosen).

to make more fine-grained domain discriminations.[6]

Singular and Plural Nouns

Figures 4 and 5 show signatures that represent singular and plural forms of the same noun, and their conditional probabilities in the terrorism and joint ventures corpora, respectively. Singular and plural words produced dramatically different correlations with relevant texts in both domains. For example, Figure 4 shows that 83.9% of the occurrences of the singular noun "assassination" appeared in relevant texts, but only 51.3% of the occurrences of the plural form "assassinations" appeared in relevant texts. Similarly, in the joint ventures domain, 100% of the occurrences of "venture between" appeared in relevant texts, but only 75% of the occurrences of "ventures between" appeared in relevant texts. And these were not isolated cases; Figures 4 and 5 show many more examples of this phenomenon.

Signature	Rel. %
<assassination, $murder$>	83.9%
<assassinations, $murder$>	51.3%
<car_bomb[7], $weapon-vehicle-bomb$>	100.0%
<car_bombs, $weapon-vehicle-bomb$>	75.0%
<corpse, $dead-body$>	100.0%
<corpses, $dead-body$>	50.0%
<disappearance, $disappearance$>	83.3%
<disappearances, $disappearance$>	22.2%
<grenade, $weapon-grenade$>	81.3%
<grenades, $weapon-grenade$>	34.1%
<murder, $murder$>	83.8%
<murders, $murder$>	56.7%

Figure 4: Singular/plural terrorism signatures

Signature	Rel. %
<tie-up, $entity-tie-up-with$>	100.0%
<tie-ups, $entity-tie-ups-with$>[8]	0.0%
<venture, $entity-venture-between$>	100.0%
<ventures, $entity-ventures-between$>	75.0%
<venture, $entity-venture-of$>	95.4%
<ventures, $entity-ventures-of$>	50.0%
<venture, $entity-venture-with$>	96.0%
<ventures, $entity-ventures-with$>	52.4%

Figure 5: Singular/plural joint ventures signatures

The reason revolves around the fact that singular nouns usually referred to a *specific* incident, while the plural nouns often referred to *general* types of incidents. For example, the word "assassination" usually referred to the assassination of a specific person or group of people, such as "the assassination of John Kennedy" or "the assassination of three diplomats." In contrast, the word "assassinations" often referred to assassinations in general, such as "there were many assassinations in 1980", or "assassinations often have political ramifications." In both domains, a text was considered to be relevant only if it referred to a specific incident of the appropriate type.[9]

Verb Forms

We also observed that different verb forms (active, passive, infinitive) behaved very differently. Figures 6 and 7 show the statistics for various verb forms in both domains. In general, passive verbs were more highly correlated with relevance than active verbs in the terrorism domain. For example, 77.8% of the occurrences of "was bombed by <X>" appeared in relevant texts but only 54.1% of the occurrences of "<X> bombed ..." appeared in relevant texts. In the MUC-4 corpus, passive verbs were most frequently used to describe terrorist events, while active verbs were equally likely to describe military events. Two possible reasons are that (1) the perpetrator is often not known in terrorist events, which makes the passive form more appropriate, and (2) the passive form connotes a sense of victimization, which news reporters might have been trying to convey.

Signature	Rel. %
<blamed, $suspected-or-accused-active$>	84.6%
<blamed, $suspected-or-accused-passive$>	33.3%
<bombed, $actor-passive-bombed-by$>	77.8%
<bombed, $actor-active-bomb$>	54.1%
<broke, $damage-active$>	80.0%
<broken, $damage-passive$>	62.5%
<burned, $arson-passive$>	100.0%
<burned, $arson-active$>	76.9%
<charged, $perpetrator-passive$>	68.4%
<charged, $perpetrator-active$>	37.5%
<left, $location-passive$>	87.5%
<left, $location-active$>	20.0%

Figure 6: Terrorism signatures with different verb forms

However, the active verb form was more highly correlated with relevant texts for the words "blamed" and "broke". Terrorists were often actively "blamed" for an incident, while all kinds of people "were blamed" or "have been blamed" for other types of things. The active form "broke" was often used to describe damage to physical targets while the passive form was often used in irrelevant phrases such as "talks were broken off", or "a group was broken up".

[6] In fact, the MUC-4 and MUC-5 corpora were constructed by applying a keyword search to large databases of news articles.

[7] CIRCUS uses a phrasal lexicon to represent important phrases as single words. The underscore indicates that the phrase "car bomb" was treated as a single lexical item.

[8] This signature only appeared once in the corpus.

[9] The relevance criteria are based on the MUC-4 and Tipster guidelines [MUC-4 Proceedings, 1992; Tipster Proceedings, 1993].

Signature	Rel. %
<assemble, $entity-active-assemble$>	87.5%
<assemble, $prod-infinitive-to-assemble$>	68.8%
<construct, $entity-active-construct$>	100.0%
<constructed, $facility-passive-constructed$>	63.6%
<form, $entity-infinitive-to-form$>	83.1%
<form, $entity-obj-active-form$>	69.2%
<put, $entity-passive-put-up-by$>	84.2%
<put, $entity-active-put-up$>	50.0%
<manufacture, $prod-infinitive-to-manufacture$>	86.7%
<manufacture, $prod-active-manufacture$>	53.8%
<manufactured, $prod-passive-manufactured$>	52.6%
<operate, $facility-active-operate$>	85.0%
<operated, $facility-passive-operated$>	66.7%
<supplied, $entity-active-supplied$>	83.3%
<supplied, $entity-passive-supplied-by$>	65.0%

Figure 7: Joint venture signatures with different verb forms

In the joint ventures domain, Figure 7 also shows significant differences in the relevancy rates of different verb forms. In most cases, active verbs were more relevant than passive verbs because active verbs often appeared in the future tense. This makes sense when describing joint venture activities because, by definition, companies are planning events in the future. For example, many texts reported that a joint venture company "will assemble" a new product, or "will construct" a new facility. In contrast, passive verbs usually represent the past tense and don't necessarily mention the actor (e.g., the company). For example, the phrase "a facility was constructed" implies that the construction has already happened and does not indicate who was responsible for the construction. Infinitive verbs were also common in this domain because companies often intend to do things as part of joint venture agreements.

Prepositions

In the next set of experiments, we investigated the role of prepositions as part of the text representation. First, we probed the joint ventures corpus[10] with joint venture keywords and computed the recall and precision rates for these words, which appear in Figure 8. For example, we retrieved all texts containing the word "consortium" and found that 69.7% of them were relevant and 3.6% of the relevant texts were retrieved. Some of the keywords achieved high recall and precision rates. For example, 88.9% of the texts containing the words "joint" and "venture" [11] were relevant. But only 73.2% of the texts containing the hyphenated word "joint-venture" were relevant. This is because the hyphenated form "joint-venture" is often used as a modifier, as in "joint-venture law" or "joint-venture proposals", where the main concept is not a specific joint venture. Figure 8 also shows much higher precision for the singular forms "venture" and "joint venture" than for the plural forms, which is consistent with our previous results for singular and plural nouns.

[10] These results are from the full joint ventures corpus of 1200 texts.

[11] Not necessarily in adjacent positions.

Words	Recall	Precision
joint, venture	93.3%	88.9%
tie-up	2.5%	84.2%
venture	95.5%	82.8%
jointly	11.0%	78.9%
joint-venture	6.4%	73.2%
consortium	3.6%	69.7%
joint, ventures	19.3%	66.7%
partnership	7.0%	64.3%
ventures	19.8%	58.8%

Figure 8: Recall and precision scores for joint venture words

But perhaps the most surprising result was that most of the keywords did not do very well. The phrase "joint venture" achieved both high recall and precision, but even this obviously important phrase produced < 90% precision. And virtually all of the other keywords achieved modest precision; only "tie-up" and "venture" achieved greater than 80% precision.

When we add prepositions to these keywords, we produce more effective indexing terms. Figure 9 shows several signatures for the joint ventures domain that represent verbs and nouns paired with different prepositions. For example, Figure 9 shows that pairing the noun "venture" with the preposition "between" produces a signature that achieves 100% precision. Similarly, pairing the word "venture" with the prepositions "with" and "by" produces signatures that achieve over 95% precision. And pairing the word "tie-up" with the preposition "with" increases precision from 84.2% to 100%. Figure 9 also shows substantially different precision rates for the same word paired with different prepositions. For example, "project between" performs much better than "project with", and "set up with" performs much better than "set up by".

Signature	Rel. %
<project, $entity-project-between$>	100.0%
<project, $entity-project-with$>	75.0%
<set, $entity-set-up-with$>	94.7%
<set, $entity-set-up-by$>	66.7%
<tie, $entity-tie-up-with$>	100.0%
<venture, $entity-venture-between$>	100.0%
<venture, $entity-venture-with$>	95.9%
<venture, $entity-venture-of$>	95.4%
<venture, $entity-venture-by$>	90.9%

Figure 9: Joint venture signatures with different prepositions

It is important to note that the signatures are generated by CIRCUS, which is a natural language processing sys-

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.