Enhancing Text Retrieval Performance using Conceptual Ontological Graph

Shady Shehata Fakhri Karray Mohamed Kamel
Department of Electrical and Computer Engineering
University of Waterloo
Waterloo, Ontario, Canada N2L 3G1
{shady, karray, mkamel}@pami.uwaterloo.ca

Abstract

Most of the data representation techniques are based on word and/or phrase analysis of the text. The statistical analysis of a term (word or phrase) frequency captures the importance of the term within a document. However, to achieve a more accurate analysis, the underlying data representation should indicate terms that capture the semantics of the text from which the importance of a term in a sentence and in the document can be derived. A new concept-based representation that relies on the analysis of the sentence semantics, rather than, the traditional analysis of the document dataset only is introduced.

The proposed conceptual ontological graph representation denotes the terms which contribute to the sentence semantics. Then, each term is chosen based on its position in the proposed representation. Lastly, the selected terms are associated to their documents as features for the purpose of indexing in the text retrieval.

Experiments using the proposed conceptual ontological graph representation in text retrieval are conducted. The evaluation of results is relied on two quality measures, the precision and the recall. Both of these quality measures improved when the newly developed representation is used to enhance the performance of the text retrieval.

1. Introduction

There has been considerable research in the area of document retrieval for over 30 years [1], dominated by the use of statistical methods to automatically match natural language user queries against data records. There has been interest in using natural language processing to enhance single term matching by adding phrases [3], yet to date natural language processing techniques have not significantly improved performance of document retrieval [2], although much effort has been expended in various attempts. The motivation and drive for using natural language processing (NLP) in document retrieval is mostly intuitive; users decide on the relevance of documents by reading and analyzing them. Thus, if a system can automate document analysis, this should help in the process of deciding on document relevance.

Usually, in information retrieval techniques, the frequency of a term (word of phrase) is computed to explore the importance of the term in the document. However, two terms can have the same frequency in a document, but one term might be contributing more to the meaning of its sentence than the other term. Thus, some terms provide the principal concepts in a sentence, and indicate the topics of a sentence. It is important to note that extracting the relations between predicates and their arguments in the same sentence has the potential for analyzing terms within a sentence. The information about who is doing what to whom clarifies the contribution of each term in a sentence to the meaning of the main topic of that sentence.

In this paper, a novel sentence-based representation, the Conceptual Ontological Graph (COG) is proposed. It captures the semantic structure of each term within a sentence and a document, rather than the term frequency within a document only. Each sentence is labeled by a semantic role labeler that defines the role of each term in a sentence. Then, all the terms are placed in the COG representation according to their contribution to the meaning of the sentence. Some terms provide the main topic of a sentence, and other terms provide more detailed information. Each term, which has a semantic role in the sentence, is called a concept. Concepts can be words or phrases and are totally dependent on the semantic structure of the sentence.

The COG representation is based on the conceptual graph theory and utilizes graph properties. The COG represents concepts mentioned in a sentence into a hierarchical manner and based on the sentence semantics. Thus, the proposed representation is used to provide a definite separation among concepts that contribute to the meaning of a sentence. These concepts are captured from their positions in the COG representation and added as features associated to their documents for the purpose of indexing in the text

Sixth IEEE International Conference on Data Mining - Workshops (ICDMW'06)
0-7695-2702-7/06 $20.00 © 2006 IEEE

retrieval. Weighting based on the matching of concepts in each document, is showed to have a more significant effect on the performance of the text retrieval due to the similarity's insensitivity to noisy terms that can lead to an incorrect similarity measure.

The results produced by the proposed representation have higher quality than those produced by a single-term weighting.

Following are the explanations of the important terms used in this paper:

- *Predicated-argument structure*: (i.e. John hits the ball). "hits" is the predicate (verb). "John" and "the ball" are the arguments of the predicate "hits",
- *Label*: A label is assigned to an argument. i.e: "John" has subject (or Agent) label. "the ball" has object (or theme) label,
- *Term*: is either an argument or a predicate. Term is also either a word or a phrase (which is a sequence of words),
- *Concept*: is a labeled term in the proposed representation.

The rest of this paper is organized as follows. Section 2 presents the thematic roles background. Section 3 introduces the conceptual ontological graph representation. The experimental results are presented in section 4. The last section summarizes and suggests future work.

2. Thematic Roles Background

The study of the roles associated with predicates is referred to a thematic role or case role analysis [7]. Thematic roles, first proposed by Gruber and Fillmore [4], are sets of categories that provide a shallow semantic language to characterize the predicate arguments.

Recently, there have been many attempts to label thematic roles in a sentence automatically. Gildea and Jurafsky [5] were the first to apply a statistical learning technique to the FrameNet database. They presented a discriminative model for determining the most probable role for a constituent, given the frame, predicator, and other features. A machine learning algorithm for shallow semantic parsing was proposed in [9]. It is an extension of the work in [5]. Their algorithm is based on using Support Vector Machines (SVM) which results in improved performance over that of earlier classifiers by Gildea and Jurafsky [5].

To the best of our knowledge, there is no research work that employs the full potential of the output of the role labeling task in representing text based on the semantics analysis of the text.

3. Conceptual Ontological Graph (COG)

The VSM does not represent any relation among the terms. Therefore, the sentences are broken down into individual components without any representation of either the sentence structure or the sentence semantic structure.

The proposed COG represents the sentence structure while maintaining the sentence semantics in the original documents. The output of the role labeling task, which are predicates and their arguments are presented as concepts with relations in the COG representation. This allows the use of more informative concept matching at the sentence-level and the document-level rather than individual word matching.

The proposed representation provides different nested levels of concepts in a hierarchical manner. These levels are constructed based on the importance of the concepts in a sentence, which makes use of analyzing the principal topics in the sentence. The hierarchal representation of the COG, provides a definite separation among concepts which contribute to the meaning of the sentence. This separation is needed to distinguish between the most general concepts and the detailed concepts in a sentence.

3.1. Conceptual Ontological Graph Structure

The COG representation is a conceptual graph $G = (C, R)$ where the concepts of the sentence, are represented as vertices (C). The relations among the concepts such as agents, objects, and actions are represented as (R). C is a set of nodes $\{c_1, c_2, ..., c_n\}$, where each node c represents a concept in the sentence or a nested conceptual graph G; and R is a set of edges $\{r_1, r_2, ..., r_m\}$, such that each edge r is the relation between an ordered pair of nodes (c_i, c_j).

A raw text document is the input to the proposed representation. Each document has well defined sentence boundaries. Each sentence in the document is labeled automatically based on the PropBank notations [8]. After running the semantic role labeler, each sentence in the document might have one or more labeled predicate argument structures. The number of generated labeled predicate argument structures is entirely dependent on the amount of information in the sentence. The sentence that has many labeled predicate argument structures includes many verbs associated with their arguments. The labeled predicate argument structures, the output of the role labeling task, are analyzed by the conceptual ontological graph.

First, the COG representation captures the concepts and the relations between concepts. Then, it presents the concepts with their relations into one sentence-based conceptual graph representation. Each node in the COG representation is either a concept node or a nested conceptual graph. If a node has a referent to a nested conceptual graph, this means that there is still detailed information about the topic of this node (concept), and is presented through concepts and their relations in the next level by a nested conceptual graph.

It is crucial to note that the contribution of the COG representation lies on its semantic-based hierarchical nature. This is due to the fact that most of the nodes in the COG representation refer to other nested conceptual graphs. This means that there are more detailed information in a sentence and represented by the nested graphs. Thus, the COG representation introduces the concepts of a sentence in a descending way. The highest nodes present the most general concepts of the sentence. The lowest nodes present the least detailed concepts mentioned in the sentence. This hierarchical manner presents different levels of depth for concept-based analysis within the sentence.

Consequently, the COG representation shares the same ontological behavior of the common ontologies regarding the semantics of the sentence. Thus, COG is considered as a semantic-based ontology in which it represents into its hierarchy levels of importance for each concept in a sentence.

3.2. Conceptual Ontological Graph Construction

The construction of the COG representation consists of the following steps:
- Determine the number of predicate argument structures for each sentence (e.g., a sentence that consists of one predicate (verb) has only one predicate argument structure).
- Present each labeled term, either argument or predicate, as concept or relation, respectively, in the conceptual graph. Thus, this graph maintains the predicate argument structure.
- Compute the amount of overlapping of the words in each term (word or phrase) including the predicate and its arguments.
- Clarify the ontological levels among the conceptual graphs, based on the number of overlapping terms in respect to the words.
- Construct the COG representation by combining the generated conceptual graphs from the second step into the COG representation.

As previously mentioned, the concepts (labeled terms) are placed in the COG representation according to the amount of overlapping between these terms with respect to the words. To present the levels of the COG hierarchy, five types of predicate argument structures are proposed and assigned to their corresponding conceptual graphs:
- *One*: There is only one generated predicate argument structure.
- *Main*: There is more than one generated predicate argument structure, and the main structure has the maximum number of terms (arguments) that refer to other terms in the rest of the predicate argument structures.
- *Container*: There is more than one generated predicate argument structure, and the container structure refers to the other arguments, and, at the same time, the container structure does not have the maximum number of referent terms.
- *Referenced*: The referenced structure has terms (predicate or arguments), referred by terms in either the main or the container structure.
- *Unreferenced*: The terms in the unreferenced structure are not referred by any other terms.

The proposed scheme creates a conceptual graph for each predicate argument structure. Each type of predicate argument structure is assigned to its corresponding conceptual graph. The COG presents the conceptual graphs as levels, which are determined according to their types.

The process for constructing the COG is achieved by the proposed algorithm *The COG Constructor*. The following declarations are used in the COG constructor algorithm:
- P: is a set of predicate argument structures $\{p_1, p_2, ..., p_n\}$, where each structure p_i consists of one or more term(s) of type T.
- T: is a set of terms $\{t_1, t_2, ..., t_n\}$, where each term t_i consists of one or more word(s) of type W.
- W: is a set of words $\{w1, w2, ..., wn\}$, where each word w_i represents a word in the term t.
- $isOverlapped(t_i, t_j)$: is a function that returns true, when $t_i \subseteq t_j$, and returns false, otherwise. If the $isOverlapped(t_i, t_j)$ function returns true, each word in the term t_j exists in the term t_i.
- $refer(p_i, p_j)$: is a function that returns true when the structure p_i refers to the structure p_j.
- $getNumberOfOverlapping(p_i, p_j)$: is a function that counts the number of overlapped words of the terms $t_i, ..., t_n$ and $t_j, ..., t_m$ in the structures p_i and p_j , respectively.

3.2.1 Algorithm: The COG Constructor

---

1. d is a new Document
2. **for** each sentence s in d **do**
3. create an empty conceptual ontological graph COG
4. **for** each p_i in s and $j = \{i + 1, ..., n\}$ **do**
5. create a conceptual graph CG_i
6. add a concept node c for an argument t to the CG_i
7. add a relation r for a predicate v to CG_i
8. **end for**
9. **if** (there is only one generated structure p) **then**
10. $CG_i.type$ = "one"
11. $COG = CG_i$
12. **end if**
13. **if** $(refer(p_i, p_j) == true)$ and $(max(getNumberOfOverlapping(p_i, p_j))$ **then**
14. $CG_i.type$ = "main"
15. $CG_j.type$ = "referenced"
16. make c in CG_i refers to CG_j
17. $COG = CG_i$
18. **end if**

19. **if** $(refer(p_i, p_j) == true)$ and $(p_i.type! = "main")$ **then**
20. $CG_i.type$ = "container"
21. $CG_j.type$ = "referenced"
22. make c in CG_i refers to CG_j
23. **end if**
24. **if** $(refer(p_i, p_j) == false)$ **then**
25. $CG_j.type$ = "unreferenced"
26. add an empty concept node c in the COG
27. make c refers CG_j
28. **end if**
29. **end for**

After running the COG constructor algorithm, the first graph has the type *main* in the COG representation. The *container* and the *referenced* conceptual graphs are nested in the *main* graph. A *referenced* conceptual graph is nested in either the *main* or the *container* graph. An external empty node is added to the COG representation to refer to the *unreferenced* conceptual graph.

For implementation and performance purposes, it is imperative to note that the COG representation maintains the identification number of each concept and each relation node, rather than, the values of the nodes. There is a hash table that includes the unique terms that appeared in each predicate argument structure. Thus, the COG is a hierarchy of the identification numbers of the terms that appear in each predicate argument structure in the sentence. This is the source of the efficiency of the representation. It is also important to note that ontology languages can not be able to capture the semantic-based nested conceptual graphs of the hierarchical structure of the sentence semantics.

The above algorithm is capable of constructing the COG representations and extracting the most important concepts for a document d in O(m) time, where m is the number of concepts in d. The following example illustrates the construction of the COG representation.

Consider the following sentence:
The availability of powerful microprocessors and improvements in the performance of networks has ***enabled*** *high performance* ***computing*** *on wide-area* ***distributed*** *systems*.

In this sentence, the semantic labeler identifies three target words, marked by bold, which are the predicates that represent the semantic structure of the meaning of the sentence. These predicates are *enabled*, *computing*, and *distributed*. Each one of these predicates has its own argument, based on the semantic labeling task.

Arguments labels[1] are numbered Arg0, Arg1, Arg2, and so on depending on the valency of the verb in sentence. The meaning of each argument label is defined relative to each verb in a lexicon of Frames Files [8].

[1]Because the meaning of each argument number is defined on a per-verb basis, there is no straightforward mapping of meaning between arguments with the same number. For example, arg2 for verb *send* is the recipient, while for verb *comb* it is the thing searched for and for verb *fill* it is the substance filling some container [8].

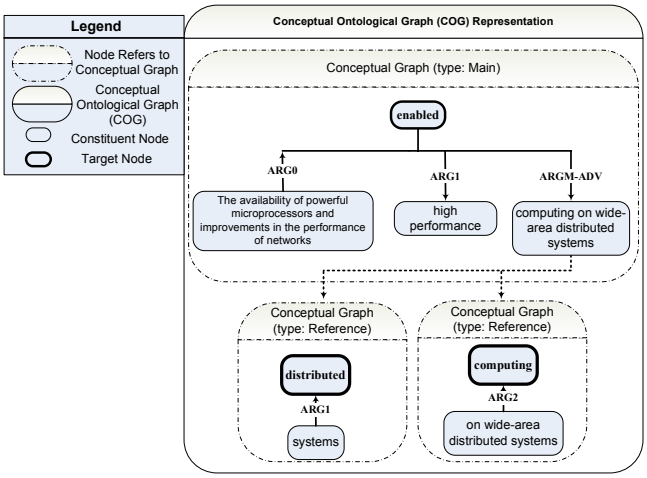

Figure 1. Conceptual Ontological Graph

The role labeler assigns the following semantic roles for each argument of each predicate:
1. For the *enabled* predicate:
(a) ARG0 is assigned to the *The availability of powerful microprocessors and improvements in the performance of networks* argument.
(b) ARG1 is assigned to the *high performance* argument.
(c) ADGM-ADV is assigned to the *computing on wide-area distributed systems* argument.
2. For the *computing* predicate:
(a) ARG2 is assigned to the *on wide-area distributed systems*
3. For the *distributed* predicate:
(a) ARG1 is assigned to the *systems* argument.

Despite this generality, Arg0 is very consistently assigned an Agent-type meaning, while Arg1 has a Patient or Theme meaning almost as consistently [8].

In this example, the first predicate argument structure of the predicate, *enabled*, consists of three arguments (or terms). The third argument, 1(c) *computing on wide-area distributed systems* has the maximum number of overlapping words to the rest of labeled terms. The term, 1(c) *computing on wide-area distributed systems* contains the other labeled terms which are *computing*, *on wide-area distributed systems*, *distributed*, and *systems*.

The predicate argument structure, which has the maximum number of overlapping words in each term, either predicates or arguments, provides the most general concepts of a sentence. In this example, the general topic of the sentence is induced by the structure of the predicate *enabled*. The topic of this example is about "X" *enables* "Y". More

detailed information about "Y" is induced from the structure of the predicate *computing* and the predicate *distributed*.

Three conceptual graphs are generated for each predicate argument structure as follows:
1. [enabled*a](ARG0?a[The availability of powerful microprocessors and improvements in the performance of networks])(ARG1?a[high performance])
(ARGM-ADV?a[computing on wide-area distributed systems])
2. (ARG2[computing][on wide-area distributed systems])
3. (ARG1[distributed][systems])

In this example, the conceptual graph of the predicate *enabled* is the most general graph that has the type *main*. The other conceptual graphs are *referenced* graphs. Thus, the COG representation of this example is illustrated in (Fig. 1).

In this example, the selected concepts, which provide detailed information about the sentence, are the concepts appear in the nested conceptual graphs. These concepts, which have the "referenced" type, are "computing", "wide-area", "distributed", and "systems".

4 Experimental Results

The experimental setup consisted of two datasets. The first data set has 12,902 documents from the Reuters 21578 dataset. There are 9,603 documents in the training set, 3,299 documents in the test set, and 8,676 documents are unused. Out of the 5 category sets, the topic category set contains 135 categories, but only 95 categories have at least one document in the training set. These 95 categories were used in the experiment as queries. The second dataset is the CRAN set of 1,398 aerodynamics abstracts with 225 queries from the Cranfield collection.

It is crucial to note that ranking algorithm is one of the most important components in a search engine because users tend to see only the first pages of the search results. Thus, the objective of the ranking algorithm is to bring documents related to the query in the first pages of the search results and show non-related documents in the last pages of the search result. In order to evaluate the quality of the ranking output, two evaluation measures the precision and the recall, which are widely used in the information retrieval literature, are adopted [6] as $P = Precision(i,j) = \frac{N_r}{T}$ and $R = Recall(i,j) = \frac{N_r}{T_r}$ where N_r is the number of relevant records retrieved, T is the total number of irrelevant and relevant records retrieved, and T_r is the total number of relevant records in the database.

The cosine[10] correlation similarity measure is adopted with the widely used term weighting scheme $TF - IDF$ (Term Frequency/Inverse Document Frequency) [11]. The cosine measure is chosen due to its wide use in the document information retrieval literature. Recall that the cosine measure calculates the cosine of the angle between the query and document vectors. Accordingly, the term-based similarity measure (sim_t) is $sim_t(q,d) = \cos(x,y) = \frac{q \cdot d}{\|q\| \ \|d\|}$ where the vectors q and d are represented as term weights calculated by using the $TF - IDF$ weighting scheme.

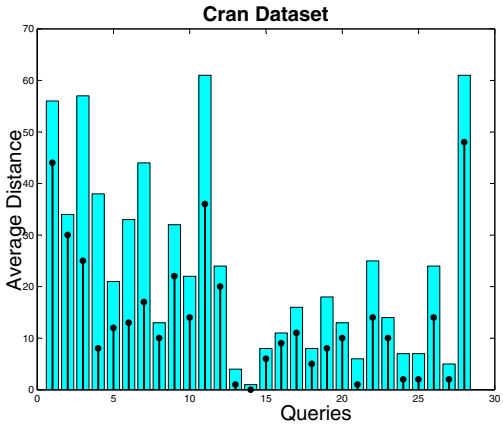

Figure 2. Search Results Ranking (Cran)

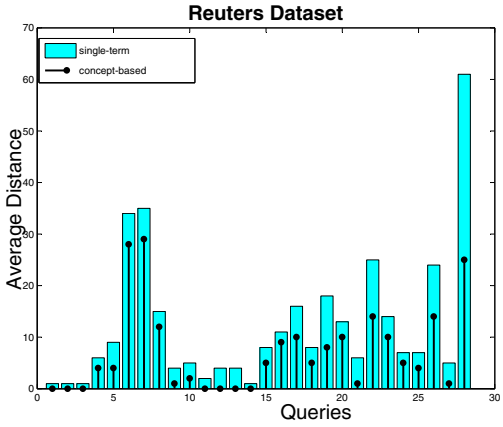

Figure 3. Search Results Ranking (Reuters)

As mentioned in section 3, The COG hierarchal nature presents the importance of the entire concepts in a sentence, providing a definite separation among concepts which contribute to the meaning of the sentence. Concepts, which have the *referenced* type, are extracted from the COG levels because they provide detailed information about a sentence. These concepts are selected and indexed as features associated to their document for the text retrieval purpose.

Each document index is associated with two features called *content* and *concept*. The *content* feature consists of the original text of the document while the *concept* feature hold the detailed concepts, which are extracted from the COG representations of the same document. For each query, the search engine searches for either concepts listed in the *concept* feature or terms appeared in the *content* feature.

It is observed that searching for concepts, rather than regular terms, increases the opportunity in retrieving docu-

Table 1. Precision of the Search Results

	Single-Term	Concept-based	Improvement
Cran	0.536	0.901	+68.09%
Reuters	0.591	0.897	+51.77%

Table 2. Recall of the Search Results

	Single-Term	Concept-based	Improvement
Cran	0.486	0.827	+70.16%
Reuters	0.452	0.841	+86.06%

ments related to the query and decreases the possibility of retrieving non-related documents. The reason behind this observation is that concepts are less sensitive to noise when it comes to calculating vectors similarity. This is due to the fact that these concepts are originally extracted by the semantic role labeler. Thus, the matching among these concepts is less likely to be found in non-related documents.

To illustrate the improvement of the of the concept-based ranking over the single term ranking, consider a ranked list of documents returned by a search engine. The distance between the position of the retrieved document, which is related to the query, in the search result and the total number of relevant documents in the database is computed.

For example, consider a query q retrieves 10 documents including document d. Although document d is related to the query q, it appears in the position number 11 in the search result. In this case, the distance $dist$ will equal to 1 as calculated by the following:
N: is the total number of relevant documents in the database
POS: is the position of document d in the search result. Thus, $dist = 0$ when $POS <= N$ and $dist = POS - N$ when $POS > N$. In Figures(2,3), the x-axis presents queries, and the y-axis presents the average distances between the positions of documents, which are relevant to the queries, in the search result and the number of documents retrieved from the database.

Basically, the intention is to minimize the $dist$ and maximize the *precision* and the *recall* to achieve high quality text retrieval. In concept-based ranking, the average distances values are much lower (lower is better) than that of the single-term weighting as illustrated in Figures(2,3).

The percentage of improvement ranges from +51.77% to +68.09% increase in the *precision* quality, and +70.16% to +86.06% increase in the *recall* quality as shown in Tables(1,2).

5. Conclusions

This work bridges the gap between natural language processing and information retrieval disciplines. A new developed representation, Conceptual Ontological Graph (COG), is proposed to improve the text retrieval performance substantially. By exploiting the semantic structure of the sentences in documents, a better performance result is achieved. This representation captures the structure of the sentence semantics represented in the COG hierarchical levels. Such a representation allows choosing concepts that actually contribute to the meaning of the sentence. This leads to perform concept matching and weighting calculations in each document in a very robust and accurate way. The quality of the ranking results achieved by this representation significantly surpasses that of traditional ranking approaches.

There are a number of suggestions to extend this work. One direction is to link the presented work to web document retrieval. Another future direction is to investigate the usage of such representations on other corpora and its effect on information retrieval results, compared to that of traditional methods.

References

[1] N. Belkin and W. Croft. Retrieval techniques. *Annual Review of Information Science and Technology*, 22:109–145, 1987.

[2] R. Cole. *Survey of the State of the Art in Human Language Technology (Studies in Natural Language Processing)*. Cambridge University Press, 1998.

[3] J. Fagan. The effectiveness of a nonsyntactic approach to automatic phrase indexing for document retrieval. *Journal of the American Society for Information Science*, 40(2):115–132, 1989.

[4] C. Fillmore. *The case for case. Chapter in: Universals in Linguistic Theory*. Holt, Rinehart and Winston, Inc., New York, 1968.

[5] D. Gildea and D. Jurafsky. Automatic labeling of semantic roles. *Computational Linguistics*, 28(3):245–288, 2002.

[6] D. Hull. Using statistical testing in the evaluation of retrieval experiments. In *Proceedings of Special Interest Group on Information Retrieval (ACM SIGIR)*, 1993.

[7] D. Jurafsky and J. H. Martin. *Speech and Language Processing*. Prentice Hall Inc., 2000.

[8] P. Kingsbury and M. Palmer. Propbank: the next level of treebank. In *Proceedings of Treebanks and Lexical Theories*, 2003.

[9] S. Pradhan, W. Ward, K. Hacioglu, J. Martin, and D. Jurafsky. Shallow semantic parsing using support vector machines. In *Proceedings of the Human Language Technology/North American Association for Computational Linguistics (HLT/NAACL)*, 2004.

[10] A. Strehl, J. Ghosh, and R. Mooney. Impact of similarity measures on web-page clustering. In *Proceedings of 17th National Conference on Artificial Intelligence: Workshop of Artificial Intelligence for Web Search (AAAI)*, pages 58–64, 2000.

[11] Y. Yanga and J. O. Pedersen. Automatic labeling of semantic roles. In *Proceedings of 14th International Conference on Machine Learning (ICML)*, pages 412–420, 1997.