An Agent-Based Approach to Dynamic Ontology Construction

Klaus Voss and Keiichi Nakata

Abstract—**The building of exhaustive ontologies leads to well known problems such as terminology, scope, encoding and context, which can only be resolved in a process of intense communication of the potential users. We propose an environment that enables users to define rules, parameters, constraints for an agent-based system which sustains (self-) organization of small sets of concepts extracted from a specific set of user provided documents and their relations. The system allows users to build or train agents, which carry small ontologies together with specific sample documents, and a generic set of rules, which enables the agents to negotiate their local ontological relations with each other.**

Index Terms—**Agents, concept classification, naïve Bayes classifier, ontology engineering.**

I. INTRODUCTION

According to John Sowa [1], "the central focus of ontology is the classification of the physical or conceptual entities;" the result is usually a form of a static map of concepts and their relations [2], or some other kind of fixed structure. The development of an actual ontology itself is a process, and since an exhaustive ontology can never be really finished [3], unless it covers a very limited domain, the developing of a static ontology is a continuous process without precisely defined termination criteria.

Due to their static nature, existing ontologies can describe widely accepted positive facts, but in return lag behind the current state of the art of the domain they try to cover. Merging and aligning existing ontologies do not necessarily accelerate the process, because only one ontology can be merged or aligned with a second one at a time, basically resulting in the re-engineering of the two initial ontologies. Thus ontologies are valuable for describing knowledge about well known facts in a machine-readable way and in particular for the interchange of this kind of knowledge between users who share the specific definition of concepts for a specific domain. In other words, whilst ontologies are useful in well-established domains to "provid[e] semantics for annotations in web pages" [3], their usefulness is limited when it comes to ongoing research, since any significant research is questioning or extending the actual state of affairs.

Klaus Voss is affiliated to Japan Science and Technology Corporation (JST).

Keiichi Nakata is affiliated to The Institute of Environmental Studies, The University of Tokyo. The authors' contact address is: The Institute of Environmental Studies, Graduate School of Frontier Sciences, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, 113-0033 Japan (phone/fax: +81-3-5841-8087, email: voss@cse.k.u-tokyo.ac.jp, nakata@k.u-tokyo.ac.jp).

Nevertheless, ontologies are used to sustain scientific research in various ways, e.g., to enhance the performance of information retrieval systems, in that they automatically add information to a query provided by a user [4], and to annotate scientific papers, so that their content can be fed into information extraction systems, which are then used to enable intelligent search in databases [5].

To summarize the usage of ontologies, it can be said, that the general issue of ontology is "how best to structure concepts for effective computation" [6]. Ontologies are used here not only to provide a shareable knowledge base, but also to facilitate their efficient utilization. Efficiency becomes crucial here, since the same experts, who are supposed to use such systems to cope with the explosion of results in many scientific domains, are needed to synchronize the ontologies in use with the state of the art in their domain. Current tools are focusing on describing a fixed structure, not on updating them to keep pace with the ongoing progress. Obviously, the gain in the efficiency provided by this structure has to be balanced with the efforts to build such a structure.

The concept of an ontology as a central allying instance for its users affords an exhaustive and reliable common language and understanding of the domain it represents, for the purposes of the whole community of users. This means it must be verified and confirmed by well-established authorities in the field, especially when it comes to describing the prevailing state of the art. However, it is very difficult to find experts who are willing to invest their time in this task, and it also requires a time-consuming process, until two or more experts agree on one ontology. The existence of a general ontology would significantly increase recall and precision of information retrieval mechanisms [4], but here is certainly a limit, where the effort of extension of the ontology exceeds the gain in performance. It is then easier to work with an information retrieval system that might have inferior performance and manually check through its results.

To make the process of ontology building more efficient, there exist tools which sustain the manual process of aligning or merging one existing ontology with another (for example [3]). These tools do not aim to find new concepts or identify their relations, which are not already included. Tools have been developed, which help to formally describe and edit ontologies (for instance [7]), but they do not sustain the *process* of finding, gathering and extracting of information and knowledge which is the prerequisite to forming concepts and arranging them in a map.

II. Limitation of Current Approaches

The building of exhaustive ontologies leads to well known problems (terminology, scope, encoding, and context) which can only be resolved in a process of intense communication of the potential users. The proposed system aims to support this essential initial part of the process of ontology *building*.

There is a certain amount of knowledge coded into an ontology, which limits the performance of any retrieval system based on this ontology. Such systems serve well to retrieve all information which is explicitly related to a certain topic (according to the information coded into the underlying ontology). This gain in efficiency (increase in recall and precision) has to be balanced with the effort to build up a comprehensive ontology, taking into account the effort imposed upon the end user to learn the ontology. Hence the effort required by experts becomes demanding, since first they must evaluate the current domain itself and there is an extra effort of having to code the knowledge they have gained into the ontology.

This ontology would then become the basis for a retrieval system, which is supposed to help other domain experts to retrieve information out of a given knowledge base. Thus, the effort of building ontology imposes limits to its level of detail, which in turn affects the performance of the retrieval system built on this ontology.

Furthermore, the later modification (due to progress in the domain) of an ontology is labor intensive, and also affects the retrieval system. Extension of a given ontology due to aligning or merging ontology still requires the workforce of domain experts and thus contributes to only gradual increases in efficiency.

III. Scope of the Proposed System

Based on observations in preceding sections, we propose a system, which does not aim to build an exhaustive global hierarchical ontology, but a network of small local sub-ontologies. The ultimate goal is to categorize an existing, finite text corpus in as much detail as necessary in the most efficient way.

For the task of categorizing, the combined information contained in the texts is used, as well as existing ontologies to automate this process as much as possible. The resulting structure of the text corpus should significantly increase the speed of the annotation of the texts in order to enhance the precision of information retrieval systems, which make use of these annotations.

The starting point is a given text corpus of one specific domain, and a shallow ontology, according to which a small portion of the corpus has been categorized and marked up. The system should help to categorize the entire text corpus, and mark up the texts automatically according to additional ontologies that already exist and are more detailed. The local ontologies are agent-based and thus provide means to identify simple relations with other ontologies (is-related, is-similar, and not-related). Here, "local" as opposed to "global" means that all relations, that are built automatically, are nearest neighbor relations only between concepts in a cluster and between two such clusters.

The user starts with limited knowledge of the domain, which he wants to explore with the help of this system. The system tries to complement the initial knowledge of the user, and recommend documents, which might extend users' knowledge. The agents should learn both from user input as well as from

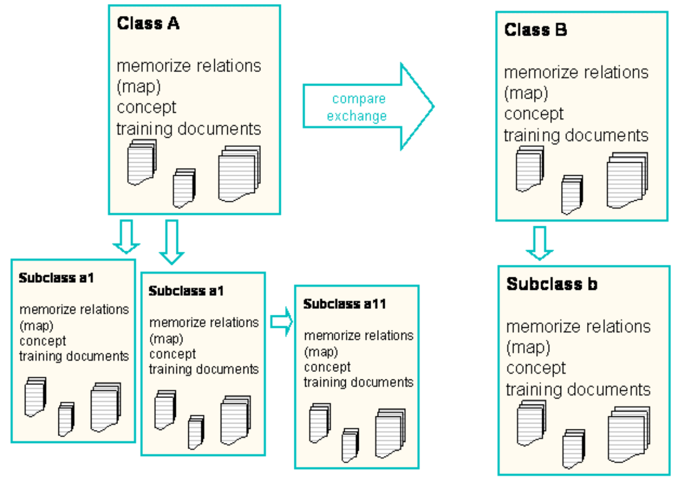

Fig.1. Schematic description of concept relations.

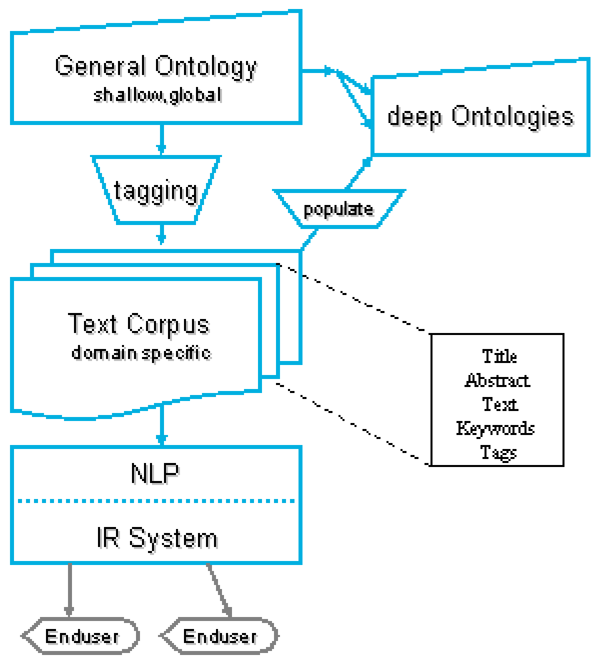

Fig 2. An example of schema for generating ontologies from texts and user interactions.

each other: The category agents try to improve themselves, in that they build relations (and memorize them) with other agents, and adjust (merge/align) their initial relations. Fig.1 illustrates the relations between concepts and local ontologies. We expect relations to be built automatically from texts. Fig. 2 describes a simplified example taken from an actual project.

The initial limited knowledge of a domain is contained in a shallow ontology, which can be developed with reasonable effort by a domain expert together with an ontology expert. Every node of the ontology represents a concept, which is populated with example texts taken form a text corpus with documents specific to the domain. Thus, the concept belonging to the node is described explicitly by a cluster of documents, which is used to train the agent which belongs to that node. The following explicit information is then available to describe a concept:

- A category name (given by the user)
- A set of documents, which explicitly describe the category (given by the user)
- A machine-readable representation of the category. A naïve Bayes classifier is a candidate, which consists of a set of keywords and their weights.
- Optional keywords, which must or must not be in a category (user input)
- Optional tags (automatically extracted) according to a given shallow ontology.

According to these initial categories, the system attempts to cluster the whole text corpus, and to assign the clusters to the given categories (e.g., provide the probability that a document belongs to a cluster). A cluster is built automatically by the system, based on the naïve Bayes algorithms with an optimization for domain specific documents [8]. For example, Weka [9] can be used to transform a set of training documents into a category to create a classifier.

As indicated above, a category is defined by the user. The system can identify the probabilities between documents and clusters, or clusters and categories. Its elements are as follows:

- *Instance*
  - *Instance* = *Data* + *Attributes*
  - *Attributes* = *Keywords* and classifier class (hit | miss)
  - *Data* = frequencies of *Keywords*
- *Instance set* (initially empty)
- Training based on example texts
  - Instances are filtered (discrete filter) and added to instance set.
- *Keywords*
  - Extracted from text, extracted from tags in the document, or provided by user (according to the ontology).

Based on the instance set, the text corpus is clustered with the help of the keywords of the deep ontology. The result is a set of clusters of the documents in the text corpus, according to the deep ontology. The naïve Bayes algorithm is modified which leads to a bias in favor of domain-specific terms [8].

Next, the clusters resulting from the deep ontology can be compared with the categories of the shallow ontology (which contains additional information to the keywords). If several clusters of the deep ontology match one category in the shallow ontology, the training documents of the category are exchanged, and the text corpus is re-clustered. Depending on the result of the re-clustering, it can be decided, if categories need to be merged or split.

Once mutually disjunctive categories are detected (based on the assignment of texts), they can be used to enhance the negative training of a category. Precision can further be enhanced with the detection of keywords, which must not be in a category.

The proposed process should facilitate the successive categorizing of a text corpus with the help of all available explicit information to minimize user input.

IV. Preliminary Analysis

In order to test the feasibility of the proposed approach, we carried out preliminary experiments on the applicability of naïve Bayes classifiers to the identification of concept relations.

First, seven concepts were selected as test cases. These were: *Murakami* (referring to the Japanese novelist Haruki Murakami), *Endo* (referring to the Japanese novelist Shusaku Endo), *Fusion* (for nuclear fusion), *Nuclear* (for nuclear power in general), *Baseball*, *Football* (for American Football), and *Soccer*. For each concept, 10 documents were collected using commercial search engine categories that correspond to these concepts. Using these documents as training documents, a naïve Bayes classifier was created for each of these concepts.

A. Construction of naïve Bayes classifiers

Since all the documents were in Japanese, they were first analyzed by a morphology processor ("Chasen"[10]) so that word boundaries and parts of speech could be identified. Then only the nouns (minus a set of meaningless words) were used in the construction of the classifier.

To test the performance of the classifier, further 5 documents were chosen for each concept, constructing a test collection of 35 documents. As a result, 33 out of 35 documents (94%) were classified correctly demonstrating that the classifiers were appropriately constructed.

B. Identification of concept relations

To find out whether naïve Bayes classifiers could be used to identify relations between concepts, the following test was conducted. In addition to the 7 concepts and 70 training documents, further 3 concepts were introduced in trained by 10 documents each to construct a naïve Bayes classifier. These concepts were: *Shakespeare* (referring to William Shakespeare), *Novel* (referring to novels in general) and *Sport* (referring to sport in general). Using these 10 concepts, we attempt to find relations among them. The relations we consider here are 1) associated (similar) and 2) class-subclass relations.

1) Measuring concept similarity

What is stored in a naïve Bayes classifier are values of conditional probability for a term *t* that appears in the class *c*. This means that internally, a set of term-probability pairs is stored. If we interpret this set of terms as elements in a vector and the conditional probability as weights, this becomes analogous to the text (term) vector representation used in text analysis [11]. Then a standard similarity measure such as cosine similarity can be used to evaluate the association between concepts.

The result is shown in Table 1. In Table 1, relations that have the cosine similarity of 0.44 and above are highlighted. This shows that the three groups of concepts (novel-related, energy-related, sport-related) are roughly identified. Based on this observation, cosine similarity of conditional probability vectors could be seen as a candidate for identifying the association relations among concepts.

2) Finding hierarchical relations

While cosine similarity could be used to identify relations that two or more concepts are somehow close to each other, it does not specify how they are related. When ontological relations are considered, the hierarchical relation (class-subclass relation) is among the most important relations. It would be therefore useful if such a relation can be automatically identified.

For this purpose, we used a set of documents which we call *probe documents*. A probe document (or a "probe") is used to see how a classifier responds to it, in this case the value of

$$-\log\{P(c_k)\times\prod_{j=1}^{d}P(x_j \mid c_k)\}$$

where $P(c_k)$ is the probability of concept c_k occurring, $P(x_j|c_k)$ is the conditional probability of term x_j occurring given concept c_k. Since classifiers that represent concepts that are similar should respond in a similar manner, probe documents could be another candidate for finding similar concepts.

The result is shown in Table 2, and cells with less than 1000 are highlighted. This also seems to indicate well, apart from the obvious exception of "Baseball" and "Novel", clusters of concepts that are similar.

If we employ a hypothesis that concepts that are more general (higher in the conceptual hierarchy) tend to be on average more likely to give favorable scores to more probe documents, taking the average of the column (or row) might reveal certain tendencies. However, upon inspection, the differences in average scores among these concepts were not significant.

We considered that the reason that this lack of difference was due to the diversity of documents used for probing. Therefore, for each concept, if we added up the score difference for only those probe documents that are given a highest score for that concept, then the difference should be more significant (Table 3).

TABLE 1.
COSINE SIMILARITY OF CONCEPTS CALCULATED FROM CLASSIFIER DATA

	Murakami	Endo	Shakespeare	Novel	Fusion	Nuclear	Baseball	Soccer	Football	Sport
Murakami	1.000	0.473	0.410	0.542	0.261	0.221	0.298	0.288	0.241	0.314
Endo	0.473	1.000	0.548	0.662	0.388	0.314	0.386	0.354	0.295	0.392
Shakespeare	0.410	0.548	1.000	0.683	0.345	0.308	0.400	0.373	0.317	0.413
Novel	0.542	0.662	0.683	1.000	0.444	0.322	0.389	0.400	0.320	0.418
Fusion	0.261	0.388	0.345	0.444	1.000	0.448	0.289	0.300	0.256	0.330
Nuclear	0.221	0.314	0.308	0.322	0.448	1.000	0.276	0.277	0.233	0.293
Baseball	0.298	0.386	0.400	0.389	0.289	0.276	1.000	0.404	0.334	0.534
Soccer	0.288	0.354	0.373	0.400	0.300	0.277	0.404	1.000	0.570	0.658
Football	0.241	0.295	0.317	0.320	0.256	0.233	0.334	0.570	1.000	0.640
Sport	0.314	0.392	0.413	0.418	0.330	0.293	0.534	0.658	0.640	1.000

TABLE 2.
SUM OF SCORES FOR PROBE DOCUMENTS

	Murakami	Endo	Shakespeare	Novel	Fusion	Nuclear	Baseball	Soccer	Football	Sport
Murakami	0	871	515	849	1918	1730	1010	1246	1492	1475
Endo	871	0	808	898	2102	1877	1337	1536	1815	1633
Shakespeare	515	808	0	892	1825	1639	712	1014	1265	1218
Novel	849	898	892	0	1881	1823	1435	1636	1896	1672
Fusion	1918	2102	1825	1881	0	765	1878	1939	2017	1961
Nuclear	1730	1877	1639	1823	765	0	1639	1701	1723	1765
Baseball	1010	1337	712	1435	1878	1639	0	713	925	776
Soccer	1246	1536	1014	1636	1939	1701	713	0	673	519
Football	1492	1815	1265	1896	2017	1723	925	673	0	574
Sport	1475	1633	1218	1672	1961	1765	776	519	574	0

TABLE 3.
SUM OF SCORES DIFFERNECE FOR PROBE DOCUMENTS THAT ARE CLASSIFIED TO EACH CONCEPT

	Murakami	Endo	Shakespeare	Novel	Fusion	Nuclear	Baseball	Soccer	Football	Sport	Average
Murakami	0	0	0	0	0	0	0	0	0	0	0
Endo	145	0	137	111	161	160	154	155	168	153	134.4
Shakespeare	52	53	0	24	91	100	65	74	88	74	62.1
Novel	36	40	55	0	70	84	80	79	91	79	61.4
Fusion	149	131	140	114	0	76	142	133	131	125	114.1
Nuclear	165	142	148	130	25	0	144	140	135	133	116.2
Baseball	98	90	84	83	103	110	0	90	93	44	79.5
Soccer	179	190	169	178	185	185	123	0	76	44	132.9
Football	227	236	210	222	212	232	185	125	0	78	172.7
Sport	64	56	61	43	20	38	39	33	28	0	38.2

From Table 3, we can observe that "Sport" has the lowest average score, and "Shakespeare" and "Novel" also display low value. Since there was no document that was classified to "Murakami", its score is 0. From this result, with the exception of "Novel", both "Sport" and "Novel" are distinguished from other concepts, and this might be able to be used as an indicator that they are super classes within each cluster of concepts.

3) Discussion

As we have shown, in our preliminary analysis, we conducted several tests in order to find strategies for identifying concept relations. It appears that we can identify association relations between concepts by finding similar concepts. Also, an indication that more general concepts could be identified, leading to hierarchical relations, has been obtained. However, these results are still preliminary and the data sets used were not large enough to make any general statements about the feasibility of these methods.

V. Conclusion

In this paper, we proposed a framework for supporting the process of ontology building, usage, extension, and maintenance. This approach does not aim to build an exhaustive global hierarchical ontology, but a network of small local sub-ontologies. The ultimate goal is to categorize an existing, finite text corpus in as much detail as necessary in the most efficient way, striking the balance between the performance of information retrieval performance and the effort of building and maintaining the ontology, which forms the basis of the retrieval system.

In this framework, each concept is represented by a naïve Bayes classifier constructed from text examples that describes it, from which concept relations and concept-cluster relations are expected to be computed semi-automatically. Concept clusters that constitute local sub-ontologies are captured by agents so that networks of sub-ontologies can be formed dynamically and global ontologies would emerge.

For this purpose, a preliminary analysis was carried out, and some strategies have been tested. Although some promising results were obtained, further tests and experiments on the feasibility and effectiveness of this approach are required.

Acknowledgment

The authors would like to acknowledge Yohei Iida, a student in the Department of Systems Innovation, School of Engineering, The University of Tokyo, for his contribution to programming and the production of preliminary results presented in this paper. This work was supported by CREST Program, Japan Science and Technology Corporation (JST).

References

[1] J. Sowa, “Knowledge Representation,” Brooks/Cole, 2000.
[2] K. Nakata, A. Voss, M. Juhnke, T. Kreifelts, “Collaborative Concept Extraction from Documents,” in U. Reimer, editor, *Proc. Second International Conference on Practical Aspects of Knowledge Management* (PAKM‘98), 1998.
[3] N. Fridman Noy, M. A. Musen:, “PROMPT: Algorithm and Tool for Automated Ontology Merging and Alignment,” in AAAI/IAAI 2000, 2000, pp.450-455.
[4] Y. Ohta, Y. Yamamoto, T. Okazaki, I. Uchiyama, T. Takagi, “Automatic Construction of Knowledge Base from Biological Papers,” *Proc. Fifth International Conference on Intelligent Systems for Molecular Biology* (ISMB’97), 1997, pp. 218-225.
[5] T. Ohta, Y. Tateisi, “Ontology Based Corpus Annotations and Tools,” in Proc. 12th Genome Informatics, 2001, pp. 469-470.
[6] F. Castel, “Ontological Computing,” Communications of the ACM, Vol. 45, No. 2, 2002, pp.29-30.
[7] S. Decker, M. Erdmann, D. Fensel, and R. Studer, “Ontobroker: Ontology based access to distributed and semi-structured information,” in R. Meersman et al., editor, DS-8: Semantic Issues in Multimedia Systems, Kluwer Academic Publisher, 1999 pp. 351-369.
[8] E. Frank, G. Paynter, “Domain-Specific Keyphrase Extraction,” IJCAI 1999
[9] I. H. Witten, E. Frank, L. Trigg, M. Hall, G. Holmes, S. J. Cunningham, “Weka: Practical Machine Learning Tools and Techniques with Java Implementations,” *Proc. ICONIP/ANZIIS/ANNES'99 Workshop on Emerging Knowledge Engineering and Connectionist-Based Information Systems*, Dunedin, New Zealand, 1999, pp. 192-196.
[10] Nara Institute of Science and Technology 1997 http://chasen.aist-nara.ac.jp/
[11] G. Salton. “Automatic Text Processin,.” Addison-Wesley Publishing, Reading, Mass., 1989.