Designing clustering methods for ontology building: The Mo'K workbench

Gilles Bisson[1], **Claire Nédellec**[2] and **Dolores Cañamero**[2]

Abstract. This paper describes Mo'K, a configurable workbench that supports the development of conceptual clustering methods for ontology building. Mo'K is intended to assist ontology developers in the exploratory process of defining the most suitable learning methods for a given task. To do so, it provides facilities for evaluation, comparison, characterization and elaboration of conceptual clustering methods. Also, the model underlying Mo'K permits a fine-grained definition of similarity measures and class construction operators, easing the tasks of method instantiation and configuration. This paper presents some experimental results that illustrate the suitability of the model to help characterize and assess the performance of different methods that learn semantic classes from parsed corpora.

1. INTRODUCTION

In this paper we propose a workbench that supports the development of conceptual clustering methods for the (semi-) automatic construction of ontologies of a conceptual hierarchy type from parsed corpora. The elaboration of any clustering method involves the definition of two main elements—a distance metrics and a classification algorithm. In the context of conceptual hierarchy formation, the Natural Language Processing (NLP) community has investigated the notion of distance to elaborate the semantic classes underlying hierarchies. Classification algorithms have been broadly studied within the Machine Learning and Data Analysis communities.

Different tools have been developed for the automatic or semi-automatic acquisition of semantic classes from "near" terms. The notion of semantic proximity is based upon distance among terms, defined as a function of the degree of similarity of the contexts. Descriptions of term contexts (the learning examples) and of the regularities to be sought vary in different approaches. Contexts can be purely graphic—words co-occurring within a window—as in the case of [1], [4]; in some cases, the window can cover the whole document (see e.g. [21]). Contexts can also be syntactic, as in the approaches that we have taken into account to develop our model, e.g. [13], [14], [11], [20], [5], [26], [7]. However, the selection of a suitable distance for a given corpus and task is still an open problem that has not received much attention so far [25]. In most cases, the criteria proposed to support this choice rely on the evaluation of the application task for which learning takes place, as described for instance in [12]. Evaluation criteria proposed to assess learning results are purely quantitative, and comparative analyses of these criteria are rare [5]. A proper characterization of the effects that different methods have on the learning results would provide methodological guidelines to help the designer select the most suitable method for a given corpus and task, or to provide support to create a new one.

This observation also applies to classification algorithms. No methodology or tool has been proposed to support the elaboration of conceptual clustering algorithms that build task-specific ontologies. Work on conceptual clustering (e.g., [19], [8], [9], [2], [1], [26]) has not been extensively applied to the problem of learning from corpora. One must however acknowledge that the application of conceptual clustering techniques to this domain is not straightforward, as existing algorithms must be previously adapted. As in the case of distances, the elaboration and selection of a suitable algorithm for a given corpus and task requires the development of new methodological guidelines and tools.

As a first step toward this goal we propose Mo'K, a configurable workbench to support the comparison, evaluation, and elaboration of methods to learn conceptual hierarchies. The conceptual clustering model underlying Mo'K permits a fine-grained definition of the components of distances and of class construction operators, easing the tasks of method instantiation and configuration. The model is extended with a set of variables that permit to characterize features specific to the elaboration of learning corpora, such as pruning, stop-lists, etc. The workbench also includes evaluation criteria to assess learning results obtained for different parameter configurations. We finally present some experimental results that illustrate the suitability of the model to help characterize different methods and assess their performance. These results concern only class formation, not classification algorithms.

2. FRAMEWORK

2.1 Learning semantic classes

In the context of learning semantic classes, learning from syntactic contexts exploits syntactic relations among words to derive semantic relations, following Harris' hypothesis [15]. According to this hypothesis, the study of syntactic regularities within a specialized corpus permits to identify syntactic schemata made out of combinations of word classes reflecting specific domain knowledge. The fact of using specialized corpora eases the learning task, given that we have to deal with a limited vocabulary with reduced polysemy, and limited syntactic variability.

[1] HELIX Project, INRIA Rhône-Alpes, ZIRST, 655 Avenue de l'Europe, F–38330 Montbonnot, email: Gilles.Bisson@imag.fr

[2] Inference and Learning Group, LRI, Bât. 490, CNRS UMR 8623 & Université de Paris-Sud, F–91405 Orsay Cedex, email: {cn, lola}@lri.fr

In syntactic approaches, learning results can be of different types, depending on the method employed. They can be distances that reflect the degree of similarity among terms [13], [26], [22], distance-based term classes elaborated with the help of nearest-neighbor methods [11], [14], degrees of membership in term classes [24], class hierarchies formed by conceptual clustering [20], or predicative schemata that use concepts to constraint selection [1], [10], [7]. The notion of distance is fundamental in all cases, as it allows to calculate the degree of proximity between two objects—terms in this case—as a function of the degree of similarity between the syntactic contexts in which they appear. Classes built by aggregation of near terms can afterwards be used for different applications, such as syntactic disambiguation [23], [24] or document retrieval [11]. Distances are however calculated using the same similarity notion in all cases, and our model relies on these studies regardless of the application task.

2.2 Conceptual Clustering

In our case, ontologies are organized as multiple hierarchies that form an acyclic graph where nodes are term categories described by intention, and links represent inclusion, seen in this case as a generality relation. Learning though hierarchical classification of a set of objects can be performed in two main ways: top-down, by incremental specialization of classes, and bottom-up, by incremental generalization. We have adopted a bottom-up approach due to its smaller algorithmic complexity, and its understandability to the user in view of an interactive validation task. In this article we focus on the elements needed to build and evaluate the basic classes of this graph, i.e. criteria for building the initial corpus, distances, and evaluation criteria to asses results. We do not address the generic class construction algorithm. With respect to this latter, let us just mention that the application of hierarchical (conceptual [19] or numerical [6]) clustering algorithms to our problem is not straightforward, given that we must build acyclic graphs with few abstraction levels, rather than deep and strict hierarchies.

3. THE MO'K MODEL AND WORKBENCH

3.1 Representation of examples and results

Following the standard practice, we use binary grammatical relations as syntactic contexts. Examples are therefore represented by triplets <Head – Grammatical Relation – Modifier head>, where <Modifier head> is the object that must be classified and <Head – Grammatical Relation> represents the attribute. The number of occurrences of a triplet in a corpus characterizes the attribute for an example. For instance, if we are interested in verbal attachments, the following two sentences:

- This causes a decrease in […].
- This high rate results from an increase in […].

allow to generate two triplets, <cause Dobj decrease > (29), and <Result from(Adj) increase > (2), both presenting the structure <Verb – grammatical relation– Head noun> (total number of occurrences of these triads in the corpus). In the remaining of the paper, we will designate by Action the tuple <Verb – grammatical relation>. Actions <Cause Dobj> and <Result from(Adj)> can be regarded as objects, and nouns <Decrease> and <Increase> can be considered as attributes with values 29 and 2, respectively.

In bottom-up clustering, couples of near objects or of objects and classes are incrementally grouped in order to form hierarchies or graphs of object classes. The standard in NLP is to use object classes (Figure 1.a) for the application task. In our experiments, we depart from this practice to compare learned classes, as we are interested in an extensional representation; we therefore use classes formed by the union of attributes of near objects (Figure 1.b).

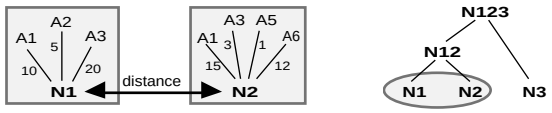

Figure 1.a. Object classes (= Nouns).

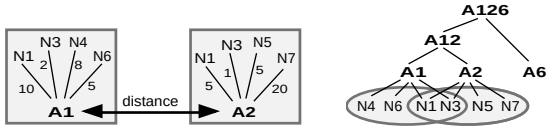

Figure 1.b. Attribute classes (= Nouns).

Let us take an example. If the objects <Cause Dobj> and <Result from(Adj)> are selected to form a class, their attribute sets are merged. Let us suppose that <Cause Dobj> is described by the nouns {**decrease**, **increase**, *modification*, *loss*, etc.}, and <Result from(Adj)> by {**decrease**, **increase**, *composition*, *evolution*, etc.}. The noun class learned will include nouns shared by both objects (in bold), and also the complementary terms (in italics); therefore, four new triplets are induced. We will then use the "attribute class" strategy, as this way the "leaf" classes that we will later evaluate will be larger than those formed using the "object class" strategy. We will not further develop here the differences between these two viewpoints, intension and extension, since this topic is out of the scope of the paper. Let us however insist on the fact that the selection of one or the other has major effects on the learning results.

3.2 Corpus parameters

The parameters used to form a learning corpus in Mo'K include, among others, selection of learning examples, level of pruning, and "cleaning" of the corpus. Let us examine the first two.

3.2.1 Selection of learning examples

One of the goals of our model is to allow the user to compare learning results as a function of the grammatical relations selected as input. Objects and syntactic contexts used in classification vary in different approaches—i.e. verbs or nouns which are considered as similar on the grounds of their shared verbal or nominal contexts, where nouns can be verb complement heads (arguments [14], or adjuncts [7]), noun complements [11] or all of them [5], [13]. None of these approaches proposes a comparative study of results based on grammatical relations chosen in the initial corpus. Our model easily allows to specify these relations. Experiments concerning verbal relations reported below (Section 4.2) illustrate this and show significant differences among results depending on the nature of objects and attributes (whether they are nouns or verbs), and on the type of corpus.

3.2.2 Pruning

A second parameter, taken into account by most existing methods and included in our model, concerns corpus pruning as a function of the number of occurrences of an element. Pruning removes occurrences that are too infrequent and therefore would cause noise, as well as those which are too frequent and do not provide any information regarding the link between an object and an attribute. The other side of the coin is

that infrequent but important cases can be removed. Our model also allows to specify the minimum number of examples characterizing an attribute and the minimal number of attributes for an example and, for each of these constraints, the minimal total number of occurrences of the triplets being considered. The experiments reported in Section 4.3 show that the level of pruning has a major impact on the results of learning, and that the optimal level depends on the corpus.

3.3 Distance Modeling

Our goal is not to cover all the possible methods that can be used to measure similarity between examples. On the contrary, our approach focuses on methods with very precise features:

- They take syntactic analysis as input;
- They do not take into account external resources (e.g. ontologies such as WordNet [18]);
- They are based on a comparison of the distribution profiles of the attributes describing the couples of object to classify.

Different methods have been proposed in the NLP literature within this framework—among others [14], [11] [13], [5] and [7]. We have developed a generic model of these methods and implemented it in Mo'K with the aim of elaborating a comparison and evaluation methodology for them. In order to come up with a generic implementation, we have identified the steps shared by all these methods, as we will see below. Mo'K is thus a workbench that implements a set of instantiable generic methods using an object-oriented representation, as opposed to the idea of a library of methods. This approach is made possible by the fact that similarity measures can in general be regarded as a comparison of the "distribution profiles" of couples of examples. This way, two objects will be considered as neighbors if the relative occurrence frequencies of each of their attributes (i.e. of the syntactic contexts) are close. Learning examples taken into account in our model can be represented by means of a contingency table. Depending on the representation hypothesis adopted, rows (examples) and columns (attributes) of the table represent different things. For example, in the experiments reported in Section 4.2 they first represent actions and nouns, respectively, and nouns and actions later on. In any case, a table cell contains the number of occurrences of an attribute for a given example. This table is obviously very sparse, as examples are generally described by a small number of attributes (see Figure 2).

In practice, computation of similarity can be decomposed into two major steps—weighting and similarity computation.

- *The weighting phase* changes every raw value of co-occurrences appearing in the contingency table by a coefficient, often normalized, which can be regarded as a weight or measure of the significance of the fact of examples and attributes co-occurring in the corpus. Its computation can entail two steps—the initialization of the weights of examples and attributes, usually according to their number of occurrences, and the calculation of a normalized weight of the relevance of each attribute for each example. Technically, this weighting phase is implemented in Mo'K using the 5 functions described (in pseudo-code) in Table 1.
- *The similarity computation phase* builds a similarity matrix between couples of examples. Similarity increases as a function of the number of shared attributes, but the way in which similarity between these distributions is calculated varies in the different approaches. In Mo'K this phase is implemented by a single function.

We thus see that, by means of 6 functions and using a few lines of code, it is possible to implement most similarity measures that follow a schema based on comparison of pairs of distribution profiles. Let us note that we do not make more specific hypotheses concerning the formal properties of measures—they can be similarities or dissimilarities, symmetrical or asymmetrical, and computed information can be of any type. This approach thus favors the comparison of existing methods, but also the elaboration of variants of these methods and even the creation of new ones. Once integrated in Mo'K, a method can access all the test and conceptual clustering resources of the system.

Name of the step	*Method*
Initialization of the weight of each example E: W(E)	Init_Weight_Example
Initialization of the weight of each attribute A: W(A)	Init_Weight_Attribute
For each example E	
For each attribute A of the example	
Calculate W(A) in the context of E	Eval_Weight_Example
Update global W(E)	Eval_Weight_Attribute
For each attribute A of the example	
Normalization of the W(A) by W(E)	Init_Similarity

Table 1. Functions implementing the weighting phase in Mo'K.

3.4 Distance evaluation

Even though our goal is the construction of hierarchies, it is interesting to evaluate the relevance of a distance metrics with respect to more simple tasks and to analyze its behavior as a function of the application domain and of the parameters of elaboration of the learning corpus. Mo'K offers different means of evaluation based on the first N couples of examples built by binary aggregation, i.e. the first N couples of examples with highest scores in the similarity matrix.

3.4.1 Measure of recall

As already mentioned in Section 3, the elaboration of a class gives rise to the induction of new triplets not observed in the initial corpus. Therefore, the evaluation process follows the classical schema of dividing the corpus in two partitions—learning and test. The former is used to build the similarity matrix according to the measure to be evaluated. The latter allows to measure the coverage rates of classes, i.e. their ability to recognize the triplets in the test set. We have adopted this evaluation task for two reasons. First, it corresponds to the elementary step in every process of bottom-up hierarchical clustering. Second, from a NLP perspective, it conforms to a disambiguation task.

3.4.2 Measure of precision

Despite its interest, the coverage measure only allows to evaluate the recall rate associated with the set N of selected classes. However, precision—a measure of the ability to avoid erroneous recognition of negative examples—is an equally important property of the metrics. In the end, a similarity measure that tends to over-generalize and describe object couples using a large number of attributes would reach high coverage rates, but produce classes that lack in meaning and precision. It is difficult to automatically solve the problem of evaluating unsupervised learning in the absence of negative examples. Given that we do not deal with annotated corpora, and we do not have negative examples, we face this problem in Mo'K by means of automatically generated (artificial) sample corpora. Following [5], we assume that examples generated this way will be negatives for the most part. Artifical examples are formed by randomly choosing an object and an attribute from the initial corpus, taking care that none of these examples appears in the learning set. We measure coverage rates on

artificial examples using learned classes. Since examples are randomly generated, some positive examples are generated as well (about 0.5% in the studied corpora). Although this rate of artificial positive examples might seem very low, it unexpectedly constitutes an important part of the artificial examples covered by learned classes—they cover between 0.5% and 2.5% artificial examples in our experiments. Hence, real precision can only be evaluated after negative examples in the artificially generated set have been computed by hand.

As we will see in the experiments reported in next section, it is interesting to measure other criteria in order to assess the relevance of a similarity measure—for example, the induction rate measuring the ratio between the number of induced triplets and the total number of triplets learned.

4. EXPERIMENTS AND RESULTS

The experiments reported here aim to illustrate Mo'K's parameterization possibilities and the impact that different parameter settings have on the learning results. These experiments make thus a case for the use of generic platforms to perform a systematic exploratory analysis in order to obtain sensible results in a given domain (corpus).

4.1 Training corpora

We have conducted experiments on two different French corpora—one contains cooking recipes gathered over the world wide web; the other, Agrovoc, contains scientific abstracts in the agricultural domain, and has been assembled by the INIST (*Institut de l'Information Scientifique et Technique*) of the CNRS. We have chosen these two corpora as they differ in generality and amount of technical terms, but are still close enough to allow for meaningful comparison of results. Both corpora have been analyzed using the same shallow parser. Only verbal relations of the form <Verb – Grammatical Relation – Noun> have been considered in our experiments. The output of syntactic parsing is highly noisy (between 30% and 50% mistakes) due to several factors such as grammatical and spelling mistakes, typos, and accentuation errors in the case of the cooking corpus. In Agrovoc, noise is mostly produced by the high number of technical terms mistaken by verbs, and of embedded noun complements which are erroneously attached to verbs. In Agrovoc, only 300 verbs are found versus 18,828 nouns. This is due to the fact that only part of the corpus has been considered—those triplets with a verb that appears in a list of verbs giving rise to nominalizations in the corpus. In the cooking corpus we find 1,181 verbs for 3,300 nouns, i.e. the ratio between nouns and verbs is, in average, divided by a factor of 20 with respect to Agrovoc. This reflects a higher specialization in this corpus. Finally, Agrovoc is three times as big as the cooking corpus (117,156 triplets for 168,287 occurrences in total).

4.2 Selection of example representation

The first experiment illustrates the importance of the choice of the object to be clustered and of the attribute which characterizes it. We have compared the classes learned by considering actions as objects and head nouns as attributes (denoted as action-based representation), with those learned using nouns as objects and actions as attributes (denoted as noun-based representation). The comparison (see Table 2) has been performed on both corpora, the other parameters remaining unchanged. We have applied Asium's distance. Pruning criteria are light—no minimum number of triplet occurrences, the minimum number of examples characterizing a given attribute has been set to 2, and the minimum number of attributes for an example to 3. We will further comment this setting in Section 4.3. For each corpus, the first 25 learned classes are evaluated. Coverage is measured on the test set, which comprises 20% of the whole corpus, and on the artificial set, which contains 50,000 triplets randomly generated (see Section 3.3). Each test has been repeated four times.

The first experiment has been conducted on Agrovoc. The classes learned in the action-based representation are twice as large as those learned in the noun-based representation. However, induction rates (number of induced triplets divided by the number of triplets learned by rote) are very similar (40% compared to 38%) and so are precision rates (45% in both cases). Precision rate represents the rate of negative examples among learned examples which cover artificial examples. The recall measured by the coverage rate of induced triplets on the test set is slightly better for the noun-based representation (5.3% compared to 4.7%) but remains quite low. This can be explained by the level of generality of what is learned: the best couples of learned nouns involved very general terms such as [technique-method], [influence-effect], in contrast with the numerous technical words of the corpus. Most of the actions characterizing these nouns concern general verbs such as [to present], [to observe], and [to report]. This is confirmed by looking at the best pairs of actions such as <to study Dobj> - <to analyze Dobj>. This explains why learned classes are of rather poor quality in both representations.

Corpus	*Learning object*	*% Induced tripl./ learned tripl.*	*Recall (test set)*	*Precision*
Agrovoc	Action	40 %	4.7 %	45 %
	Nom	38 %	5.3 %	45 %
Cooking	Action	34 %	12 %	32 %
	Nom	38 %	9.1 %	52 %

Table 2. Experimentation about example representation

The experiments on the cooking corpus have built classes of similar size in both representations. Induction rate is slightly higher for the noun-based representation—38% compared to 34%. However, recall on the test set is better in the action-based representation, (12% and 9.1%, respectively). Induced triplets are thus more useful in the case of action-based representation, even though they are less numerous. Moreover, the precision measured by the rate of negative artificial examples covered by learned examples is much better for the action-based representation (32% compared to 52%). Precision and recall rates are thus better in this representation, although the rate of induced triplets is smaller than in the noun-based representation. In any case, all rates are much better than the ones computed for the Agrovoc experiments. A closer examination of the best pairs of actions and nouns confirms the idea that overgeneralization is less of a problem here than in Agrovoc. Noun pairs are more precise (e.g., [fridge-freezer], [olive oil–oil]) and described by more technical actions. In the same way, the best pairs of actions, such as [absorb Dobj, evaporate Dobj], are characterized by nouns (in this case [vinegar, water, wine, excess, etc.]) which are significantly more specific than in Agrovoc. The smaller variability of the cooking corpus explains these observations, showing that the larger size of Agrovoc does not improve the meaningfulness of the regularities observed.

This experiment thus shows that the choice of a representation can have a major impact on the learning results. It is therefore advisable to select a suitable representation before addressing a new domain.

4.3 Pruning parameters

To illustrate the importance of pruning, two pruning settings have been applied to both corpora and compared. In both cases we have used Asium's distance, objects are actions and attributes are nouns. The first setting is the one described in the previous section. In the second one, we have set the minimum number of occurrences for an example (triplet) to 2, in order to remove triplets occurring only once in the corpus, as they may represent noise. We have also augmented the values of the minimal number of examples that must characterize an attribute (from 2 to 3), and of the minimal number of attributes per object (from 3 to 5, i.e. in this version the objects being compared appear in at least 5 different syntactic contexts). As it can be inferred from the histogram in Fig. 2, this setting excludes 80% of the corpus, versus 70% in the first setting. We can thus hope for a more reliable classification, to the risk of removing so many examples that coverage (recall) is drastically affected. The experiments show that, for the cooking corpus, induction rate (32%) and coverage (11.2%) on the whole test set are nearly unaffected. On the contrary, the rate of artificial triplets covered scales by a factor of 3 with respect to the previous rate; we think that this significant increase indicates that this pruning rate increases the rate of erroneously induced examples. In Agrovoc, induction rate decreases by a factor of three and coverage rate by a factor of two, whereas the rate of artificial triplets covered scales by a factor superior to 2. Recall is therefore much strongly affected by pruning than in the cooking corpus. In both cases, this new pruning setting gives rise to a decrease in performance.

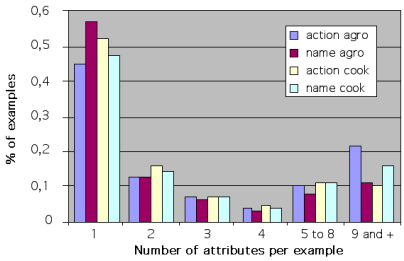

Figure 2. Example distribution per number of attribute

However, this conclusion drawn from the evaluation of basic classes must be tempered in the case of hierarchy formation. In this case, the more constraining version of pruning allows to eliminate many non-significant classes that result from the presence of closely similar actions described by a small number of attributes. It seems clear that, in a process of hierarchical clustering, this type of class would cause problems, as it would alter some groupings. Therefore, the type of pruning to be applied partly depends on the task to be performed.

4.4 Comparison of methods

This last experiment illustrates some aspects of the use of Mo'K to compare results obtained with different distances. Among the methods that we have tested with Mo'K (such as those proposed by Dagan *et al*., Hindle, Grefenstette, and Grishman *et al*. among the best known in the literature, as well as other distances proposed by the authors, e.g. Asium, and Greedy) we have chosen to compare here the distances used in Asium [7], the one proposed by Dagan [5], and Greedy[3]. The reason for this choice relies on the fact that these three methods have shown good performance when compared with others, while presenting rather different behaviors. They are thus a representative sample of the subset of methods that we have modeled as characterized in Section 3.3. These methods have been applied to the corpus of cooking recipes. Learning parameters are the same as those described in Section 4.2, objects are actions and attributes are names. In addition, we have tested the influence on learning of the number of disjointed classes on which recall performance is evaluated. To do this, we have varied this number (in abscissa of the diagrams below) between 10 and 200.

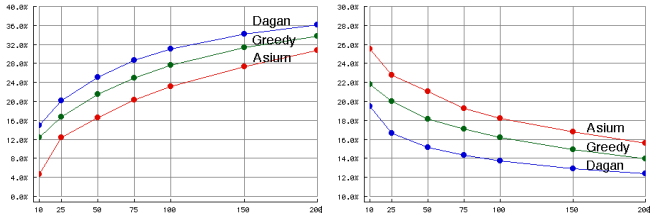

Figure 4a&b. Recall rate and Class efficiency

These diagrams show, respectively, the recall rate of each method on the test set (Figure 4a), and the efficiency of classes rate (Figure 4b). The latter is assessed by the ratio between the number of triplets learned (by rote and induced), and the number of triplets effectively used in the recall test. As we can see in the first diagram, the coverage rate of the three methods grows as expected according to the number of classes considered, Dagan's method yielding the best results. On the contrary, if we pay attention to the efficiency of classes, Asium takes better advantage of the triplets learned. Looking at both diagrams, we can conclude that these methods have different behaviors. Dagan's gives rise to more general classes (more triplets are learned, but the number of useless ones is higher). Asium constructs more specific classes (fewer triplets learned, but more of them are useful). We can take a closer look at the behavior of these methods and study the quality of induction in terms of the rate of induced triplets which are effectively used in the recall test (Figure 5).

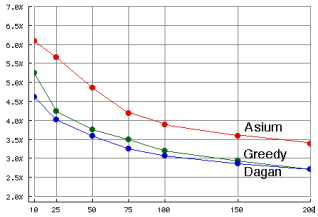

Figure 5. Quality of induction.

While the previous conclusion is confirmed for Asium and Dagan's methods, we can also note that Greedy and Dagan's methods have the same behavior along this criterion. In fact, it seems that Dagan's method is able to induce more useful triplets than Greedy, whereas this latter tends to learn by rote a more representative sample subset of the corpus.

The tests performed on the artificial examples confirm these results. Therefore, it seems that the classes learned by Dagan's method and, to a lesser extent, by Greedy are less robust and present lower precision rates for of the learning parameters chosen. We must emphasize that these conclusions *only apply to the recipe corpus*. For Agrovoc, results are considerably different. These experiments therefore show the importance of going through an exploratory process in order to come up with the most suitable methods and representation.

[3] Greedy has a simple behavior: it is based on a measure, inspired from the χ^2, which favors the selection of pairs of examples described by a large number of attributes (the method is named after this feature).

5. CONCLUSIONS AND PROSPECTS

Mo'K is a configurable workbench that supports the development of conceptual clustering methods for specific ontology building. The learning model proposed here takes parsed corpora as input. No additional (terminological or semantic) knowledge is used for labeling the input, guiding learning, or validating the learning results. Preliminary experiments showed that the quality of learning decreases with the generality of the corpus. This makes somehow unrealistic the use of general ontologies for guiding such learning, as they seem too incomplete and polysemic to allow for efficient learning in specific domains. For example, [16] points out that 40% of the words in canonical form in the titles and abstracts of the Communications of the ACM are not included in the LDOCE (Longman Dictionary of Contemporary English). This problem posed in the case of learning specific ontologies obviously does not apply in the case of guiding learning of general semantic classes, as shown in the abundant literature on the topic (see e.g. [23], [22], [24], [17]). It would however be highly valuable to take advantage of existing ontologies to improve the quality of learning. We consider that this can be achieved in two main ways. First, learning could be improved by the use of specific terminologies, dictionaries and nomenclature, such as SNOMED International in the medical domain [3]. Second, some methodological guidelines would be needed to integrate specialized learned ontologies into more general ontologies such as WordNet [18].

Although we have focused on a disambiguation-based task, other validation tasks could be integrated in Mo'K, such as query extension and information extraction. Learning specialized ontologies of high quality for these tasks will allow the development of applications in technical and rapidly evolving domains, in which manual acquisition is too costly. In this sense, we have started exploring information extraction from molecular biology abstracts.

ACKNOWLEDGEMENTS

We are grateful to INIST-CNRS for providing the Agrovoc corpus. This research is partly funded by the French Ministry of Industry under RNRT project Astuxe.

REFERENCES

[1] Basili R., Pazienza M. T. and Velardi P. 1996. An empirical symbolic approach to natural language processing. *Artificial Intelligence Journal* 85, pp. 59-99.

[2] Bisson G. 1992. Conceptual Clustering in a First Order Logic Representation. In *Proceedings of 10th European Conference on Artificial Intelligence* (ECAI'92), pp. 458-462, Vienna.

[3] Bouaud J., Habert B., Nazarenko A. and Zweigenbaum P. 1997. Regroupements issus de dépendancess syntaxiques en corpus : catégorisation et confontation à deux modélisations conceptuelles. In *Actes des Journées Ingénierie des Connaissances*, Zacklad E. (Ed.), pp. 207-223, Roscoff, France, May.

[4] Church K. W. and Hanks P. 1989. Word Association Norms, Mutual Information, and Lexicography, in *Proc. of the 27th Annual Meeting of the Association for Computational Linguistics*, pp. 76-83.

[5] Dagan I., Pereira F., and Lee L. 1994. Similarity-Based Estimation of Word Co-occurrence Probabilities. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics, ACL'94*, New Mexico State University, June.

[6] Day W., Edelsbrunner H. 1984. Efficient Algorithms for Agglomerative Hierarchical Clustering Methods, *Journal of Classification*. Volume 1. pp. 1-24.

[7] Faure D. and Nédellec C. 1998. A Corpus-based Conceptual Clustering Method for Verb Frames and Ontology Acquisition. In P. Velardi (Ed.), *Adapting lexical and corpus resources to sublanguages and applications, Workshop* of the 1st *Intl. Conf. on Language Resources and Evaluation*, pp. 1-8, Granada, Spain, May.

[8] Fisher D.H. 1987. Knowledge Acquisition via Incremental Conceptual Clustering, *Machine Learning Journal* 2, pp. 139-172.

[9] Gennari J., Langley, P., Fisher D. 1989. Model of Incremental Concept Formation, *Artificial Intelligence Journal*, Volume 40, 11-61

[10] Gomez F. 1998. Linking WordNet Verb Classes to Semantic Interpretation. In Proceedings of the *COLING-ACL Workshop on the Usage of WordNet in NLP Systems*.

[11] Grefenstette G. 1992. Use of syntactic Context to Produce Term Association Lists for Text Retrieval. In *Proceedings of the 15th International SIGIR'92*, Denmark.

[12] Grefenstette G. 1993. Evaluation Techniques for Automatic Semantic Extraction: Comparing Syntactic and Window Based Approaches. In *Workshop on Acquisition of Lexical Knowledge form Text*, Columbus, OH, June.

[13] Grishman R., Sterling J. 1994. Generalizing Automaticaly Generated Selectional Patterns. In *Proceedings of the 16th International Conference on Computational Linguistics (COLING'94)*.

[14] Hindle D. 1990. Noun classification from predicate-argument structure. In *Proc. of the 28th Annual Meeting of the Association for Computational Linguistics (ACL'90)*, pp. 268-275, Pittsburgh.

[15] Harris Z., Gottfried M., Ryckman T., Mattick Jr P., Daladier A., Harris T. and Harris S. 1989. The form of Information in Science. In *Analysis of Immunology Sublanguages*, vol. 104 of *Boston Studies in the Philosophy of Science*. Dordrecht, the Netherlands, Kluwer Academic Publishers.

[16] Krovetz R and Croft W.B., W. 1991. Lexical Ambiguity and Information Retrieval. In *Lexical Acquisition: exploiting on-line resources to build a lexicon*¿ Zernik (Ed.), pp. 45-65, Hillsdale, New Jersey, Lawrence Erlbaum Associates.

[17] Li H. and Abe N. 1998. Word clustering and disambiguation based on co-occurrence data. In *Proceedings of COLING - ACL'98*.

[18] Miller G. 1990. WordNet: an on-line lexical database, *International Journal of Lexicography*, 3(4).

[19] Michalski R.S., Stepp E. 1983. Learning from Observation: Conceptual Clustering. In *Machine Learning I: an Artificial Intelligence Approach*, Tioga, pp. 331-363.

[20] Pereira F., Tishby N. and Lee L. 1993. Distributional clustering of English words. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics ACL'93*, p. 183-190.

[21] Qiu Y. and Frei H. P. 1993. Concept based Query Expansion. In *Proceedings of 16th Annual International ACM SIGIR Conference*, pp. 160-169, Pittsburgh, ACM Press.

[22] Resnik P. 1995. Using Information Content to evaluate Semantic Similarity in a Taxonomy, *Cognitive Modelling*.

[23] Resnik P. and Hearst M. A. 1993. Structural Ambiguity and Conceptual Relations. In *Proc. Workshop on Very Large Corpora: Academic and Industrial Perspectives*, pp. 58-64, Ohio State Univ.

[24] Ribas F. 1995. On Learning More Appropriate Selectional Restrictions. In *Proceedings of EACL'95*.

[25] Roland D. and Jurafsky D. 1998. How Verb Subcategorization Frequencies Are Affected By Corpus Choice. In *Proceedings of the Int'l Conf. Computational Linguistics (COLING'98)*.

[26] Sekine S., Caroll J. J., Ananiadou S. and Tsujii J. 1992. Automatic Learning for Semantic Collocation. In *Proc. of the 3rd Conference on Applied Natural Language Processing*, pp. 104-109.

[27] Sparck Jones K. and Barber E. B. 1971. What makes an automatic kyewords classification effective?, *Journal of the ASIS*, 18: 166-175.

[28] Talavera L. and Bejar J. 1998. Efficient construction of comprehensible hierarchical clusterings. In *Proceedings of the 2nd European Symposium on Principles of Data Mining and Knowledge Discovery, PKDD'98*. pp. 93-101. Nantes, France. J. M. Zytkow and M. Quafafou (eds.) LNAI vol. 1510, Springer Verlag.

[29] Vasco J. J. F., Faicher C., Chouraqui, E. 1996. A knowledge acquisition tool for multi-perspective concept formation. In proceedings of *9th European Knowledge Acquisition Workshop, EKAW'96*, pp. 227-244. Springer Verlag.