

**Discovering Building Blocks for Human Based
Genetic Algorithms**

**Takaoki Ueda Noriko Imafuji Yasui Xavier
Llorà
Kumara Sastry David E. Goldberg
IlliGAL Report No. 2007020
□**

Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
117 Transportation Building
104 S. Mathews Avenue Urbana, IL 61801
Office: (217) 333-2346
Fax: (217) 244-5705

Discovering Building Blocks for Human Based Genetic Algorithms

Takaoki Ueda Noriko Imafuji Yasui Xavier Llorà
Kumara Sastry David E. Goldberg

Illinois Genetic Algorithms Laboratory
University of Illinois at Urbana-Champaign
104 S. Mathews Ave., Urbana, IL 61801
{tueda2, niyasui, xllora, ksastry, deg}@uiuc.edu

Abstract

The push for rapid innovation and creativity in this Internet age places a premium on effective integration of both human and computer-generated knowledge. One of the key components of a distributed and scalable environment for accomplishing this integration called DISCUS is the human-based genetic algorithm (HBGA)—a GA where humans perform genetic operations. This paper takes the first step towards designing a *competent* HBGA, which can enable humans to innovate quickly, reliably, and accurately. Specifically, this paper proposes a methodology for discovering building blocks from text documents including reports, chat, transcripts and e-mail. The proposed method has been applied to simple test problems and to a news article set. The results show that the proposed BB-identification methodology is effective and enables humans to effectively exchange the BBs for rapid innovation.

1 Introduction

Due to the rapid growth of Internet, the amount of interactions among people has exponentially increased. Therefore, rapid visualization and representation of key points of discussions or any other form of documents is critical for successful brainstorming/innovation. To address these difficult problems, DISCUS (6) a project to “*create a distributed and scalable environment for the integration of both human- and computer-generated knowledge in uncertain settings*” has been started. One of the important components of DISCUS is the human based genetic algorithm, where humans perform “judgment creative potential of participants” in the form of the evaluation and the GA operations to obtain innovative solutions during brainstorming (9).

However, most documents and discussions are disorganized; the identification of key “building blocks” is difficult. Moreover, there is no absolute or relative quality measure of conclusions drawn from discussions or documents. Elsewhere, 4) has been pointed out that the primary steps towards a successful GA design involves BB identification, and HBGA is no different. Therefore, the first step towards designing a *competent* HBGA is to develop a method that can automatically identify BBs which can be used to *innovation*, quickly, reliably, and accurately (4).

The purpose of this paper is to develop a method for discovering BBs from text-based discussions or documents. In this paper, BB discovery is analogous to discovering clusters of words in a sentence such that the set of word in each cluster indicate semantic of the sentence. Once the BBs are identified they can be used for both *innovation* (recombination) as well as *improvement* (mutation).

The remainder of this paper is organized as follows. We start with a brief description of related work in Section 2. In Section 3, we give an overview of our approach we propose for identifying BBs, followed by the details of each process of the proposed approach in Section 4-6. Section 7, presents experimental results obtained by applying our approach to simple text and a news article set. Followed by a note on future work, and key conclusions of the work is provided in Section 8.

2 Related Research

Finding building block is an important step toward designing competent GAs(4). Human-based genetic algorithms (HBGAs) (9) are first step toward modeling human innovation and creativity by means of computational embodiments. Despite HBGAs outsource GA evaluations and evolutionary operators to human beings, proper model building and linkage identification can help improving the scalability of such methods. Such situation usually leads to a need to be able to process streams of text—the normal encoding used by human communications. Therefore, we need to be able to identify building blocks that are defined based on the context and the semantics of the ongoing discourse under analysis.

GAs has been applied to a plethora of applications related to text processing, mining, and analysis. Those application include categorizing language independent multi-class text (10), inducing relevant features for automatic word classification tasks (8), and solving ambiguous labeling problems (13) etc. Another group of applications have focused on methodology for document clustering, using some sort of evolutionary inspired approaches for instance. Some of these include collection clustering using divisive clustering (2) and agglomerative clustering algorithm (1; 18). However, to the best of our knowledge few or null attempts have tried to use evolutionary approaches to discover context-aware semantic building blocks in streams of text.

A key component of our work is the intensive usage of semantic networks. In this paper we have heavily relied on WordNet to get that semantic network. WordNet is a readable lexical database developed at Princeton(11). Wordnet is a network which models the relationships between words, for example, synonyms, antonyms, hyponyms, and hypernims. It has large database of words since psycholinguistic research requires large sample of the words. WordNet accepts various usages. One of the few related works we could found is an intelligent information news filtering system (12) which uses both GAs and WordNet to fulfill its purpose.

3 Approach Overview

Our goal is to identify sets of words from given text stream that in such given particular context form semantic unity. In this section, we present a brief overview of the overall approach to facilitate its comprehension.

Our approach consists of three processes:

1. Text processing of text streams
2. Semantic groups identification
3. Context-aware semantic building blocks discovery

For example, consider the following document, where each paragraph is identified by a unique number. All of these paragraphs have similar meaning when used in the same context.

1. This is a long story which cannot be described in one paragraph. However, we can describe this story by repeating paragraphs with same meaning many times.
2. This story is ambiguous and huge. Therefore, we cannot write it in one paragraph. One way to describe this story is to repeat paragraphs using similar words.
3. There is a story which cannot be described in one paragraph. However, we can write this paragraph repeatedly to describe it. Repeating is the only acceptable solution towards expression.
4. This is a story which cannot be expressed in one paragraph. There is one way to describe it and that is by repeating paragraphs with the same meaning as this paragraph many times.
5. One paragraph is not sufficient to express our story. By writing paragraphs with same meaning many times, it is possible to express our story.
6. Without repeating paragraphs with same meaning using similar words, we can not describe our story.

Text Processing: As a first step, we need to clean the document for enhancing efficiency of later processes. This step includes stop words elimination and word stemming. The former is to eliminate words irrelevant to document context, and the latter is to normalize each word to its base form. For example, this step converts the paragraph 1 in the sample document above to “‘story describe one paragraph. describe story repeat paragraph mean time’”.

Semantic Groups Identification: A semantic group indicates a cluster of words with similar meanings. Similarities between words are measured based on *hypernyms*. A *hypernym* is a word that is more generic than a given word. Our experiments use WordNet for obtaining hypernyms of each word. Note that every word in WordNet has at least one hypernym, that is, *entity*. For example, you can find semantic groups obtained from the sample document in the Section 7. Identified semantic groups are indexed by a word in the group.

Context-aware Semantic Building Block Identification: By the previous step, we have a set of words (including index words to semantic groups). Last step is to identify building blocks from the set of words. For example, as we report in the Section 7, obtained building blocks from the sample document were “one paragraph” and “describe repeating”.

4 Text Processing

Text processing is executed prior to other processes, in order to reduce ambiguousness of text and help to create a compact word set and facilitate later processes. This section describes three text processing used in our approach: stop words elimination, stemming, and splitting.

4.1 Stop Words Elimination

A *stop word* is a word which is commonly and frequently used in texts. It is filtered out from texts, because typically, it is irrelevant to essence of texts. Our stop word list is based includes pronouns, prepositions, articles, adverbs and some verbs (20).

4.2 Stemming

Stemming is a process for normalizing variants of words by removing morphological and inflectional endings from the words. For example, by the stemming process, all of the words *remove*, *removes*, *removed*, and *removing* are converted into *remov*. Our stemming process uses Porter stemming algorithm (14), which is a widely used stemming algorithm for English. The algorithm detail is beyond the scope of this paper. Stemmed words are used internally further processes, and original words are used to show results.

5 Semantic Group Identification

A semantic group indicates a cluster of words with similar meanings. We use a design structure matrix (DSM) clustering method for identifying the semantic group. In this section, first we give a brief review of DSM clustering method. Next, the approach for identifying semantic groups by DSM clustering method is proposed.

5.1 DSM Clustering

A dependency structure matrix (DSM) is essentially an adjacency matrix representation of a graph where each entry d_{ij} represents the dependency between node i and node j (19; 21). Entries d_{ij} can be real numbers or integers. The larger the d_{ij} is, the higher the interaction is between node i and node j . If we focus on the 0-1 domain, then $d_{ij} = 0$ means that node i and node j do not interact, and $d_{ij} = 1$ means that node i and node j interact with each other. The diagonal entries (d_{ii}) have no significance and are usually set to zero or blacked-out. For elaborate exposition of DSM, please see MIT DSM web site: <http://www.dsmweb.org/>.

The goal of DSM clustering is to find subsets of DSM elements (*i.e.*, clusters) so that nodes within a cluster are maximally interacting, and clusters are minimally interacting. For In a typical DSM clustering problem, overlapping clusters (clusters that share same nodes) are permissible.

Our approach uses DSM clustering metric based on the minimal description length principle (MDL) (15). Previous DSM clustering algorithms can be found elsewhere (3; 17). Their results showed that the objective function used was short of accurately predicting “good” clustering because of the oversimplified objective function.

Suppose that we have a model which describes a given data set, $DSM = [d_{ij}]$. Here, the model means a description that specifies which node belongs to which cluster. Usually, the model does not completely describe the given data; otherwise, the model would be too complex to use. Therefore, the description length that the model needs to describe the given data consists of two parts: the model description and the mismatched data description. This scheme may be easier to understand in light of the following sender-receiver example.

Assume a sender has a given data set which is needed by the receiver. Given a model that approximately describes the given data set, the sender first sends the model (*i.e.*, model description) to the receiver. To ensure that the receiver gets exactly the same data set, the sender is also required to send the data which is mis-described (*i.e.*, mismatched data description) by the model

sent earlier. If the model is too simple, the model description is short; but many data mismatches exist and the mismatched data description becomes longer. On the other hand, a complicated model reduces mismatched data, but the model description is longer.

The minimum description length principle (MDL) (15) satisfies our needs for dealing with the above trade-off. The MDL can be interpreted as follows: among all possible models, choose the model that uses the minimal length for describing a given data set (that is, model description length plus mismatched data description length). There are two key points that should be noted when MDL is used: (1) the encoding should be uniquely decodable, and (2) the length of encoding should reflect the complexity. For example, the encoding of a complicated model should be longer than that of a simple model. Next, we define the MDL clustering metric in detail.

Model Encoding. The way we encode the model is straightforward. The description of each cluster starts with a number which is sequentially assigned to each cluster, and then this is followed by a sequence of nodes in the cluster. It is easily seen that the length of this model description is as follows:

$$\sum_{i=1}^{n_c} (\log_2 n_n + cl_i \cdot \log_2 n_n), \quad (1)$$

where n_c is the number of clusters in the model, n_n is the number of nodes, cl_i is the number of nodes in the i -th cluster.

Mismatched Data Description. Based on the model, we first construct another DSM (call it $DSM' = [d'_{ij}]$), where each entry d'_{ij} is 1 if and only if some cluster contains both node i and node j simultaneously.

Then, we compare DSM' with the given DSM . For every mismatched entry, where $d'_{ij} \neq d_{ij}$, we need a description to indicate where the mismatch occurred (i and j) and one additional bit to indicate whether the mismatch is zero-to-one or one-to-zero. Define a mismatch set $S = \{(i, j) | d'_{ij} \neq d_{ij}\}$. The mismatched data description length is given by:

$$\sum_{(i,j) \in S} (\log n_n + \log n_n + 1). \quad (2)$$

The first $\log n_n$ in the bracket indicates i , the second one indicates j , and the additional one bit indicates the type of mismatch.

The MDL clustering metric is given by the summation of the model description length and the mismatched data description. With some arithmetic manipulations, the metric can be expressed as follows:

$$f_{DSM}(M) = \log n_n \sum_{i=1}^{n_c} (cl_i + 1) + |S|(2 \log n_n + 1), \quad (3)$$

where n_c is the number of clusters, n_n is the number of nodes in the DSM, cl_i is the size of the i -th cluster, and S is a mismatch set.

With the above metric, the DSM clustering problem is converted into an optimization problem: Given a DSM, the objective is to find a DSM clustering arrangement (model, M) to minimize the above metric (f_{DSM}).

5.2 Semantic Groups Identification by DSM Clustering

Semantic groups are identified by DSM clustering method reviewed in the above. DSM clustering is mainly used because its ability to infer overlapping clusters—a common property of text where one word may have different meanings and, hence, belong to different groups. This subsection focuses on how to create DSM from a set of words.

Let W be a set of words contained in texts. Hypernym induced vectors $H(w_i)$ and $H(w_j)$ for $w_i \in W$ and $w_j \in W$ are obtained by hypernym sets of w_i and w_j . Let H_{w_i} and H_{w_j} be hypernym sets of word w_i and word w_j , respectively. Let $H = H_{w_i} \cup H_{w_j} = \{h_1, h_2, \dots, h_n\}$ denote a combined set of hypernyms derived from w_i and w_j . $H(w_i)$ and $H(w_j)$ are defined by

$$H(w_i) = \{H_{w_i}(h_1), H_{w_i}(h_2), \dots, H_{w_i}(h_n)\} \quad (4)$$

$$H(w_j) = \{H_{w_j}(h_1), H_{w_j}(h_2), \dots, H_{w_j}(h_n)\}, \quad (5)$$

where

$$H_{w_i}(h_k) = \begin{cases} 0 & \text{if } h_k \notin H_{w_i} \\ 1 & \text{if } h_k \in H_{w_i} \end{cases} \quad (6)$$

Likewise, $H_{w_j}(h_k)$ is 0 (if $h_k \notin H_{w_j}$) or 1 (if $h_k \in H_{w_j}$).

Let $\theta(w_i, w_j)$ be similarity between w_i , and w_j . $\theta(w_i, w_j)$ is measured by cosine similarity of hypernym induced vectors, as follows.

$$\theta(w_i, w_j) = \arccos \left[\frac{(H(w_i))^T \cdot H(w_j)}{\|H(w_i)\| \|H(w_j)\|} \right], \quad (7)$$

Let D denote a DSM for identifying semantic BBs is a n -square matrix, where $n = |W|$. (i, j) entry ($i \neq j$) of D is defined as follows.

$$D(i, j) = \begin{cases} 0 & \text{if } \theta(w_i, w_j) < \delta \\ 1 & \text{if } \theta(w_i, w_j) \geq \delta \end{cases} \quad (8)$$

where $w_i, w_j \in W$ and δ is a threshold. Intuitively, the entry is 0, if the corresponding two words have similar meaning, and vice versa. In our experiment, $\delta = 1.10$.

5.3 Simple Example

Assume we have a set of words $W = \{dog, cat, potato, rat\}$ derived from a given text. For example, $\theta(dog, cat) = 1.13$, then (dog, cat) entry of DSM = 1. Similarly, obtaining θ for every pair of words, DSM for W is represented as Table1.

Table 1: DSM matrix.

-	dog	cat	potato	rat
dog	x	1	0	1
cat	1	x	0	1
potato	0	0	x	0
rat	1	1	0	x

Table 2 shows the reordered matrix obtained by DSM clustering.

From this matrix, we have a cluster with “dog”, “cat” and “rat” and a singleton “potato”.

6 Context-aware Semantic Building Blocks Identification

A context-aware semantic building block indicates a cluster of words having close correlations, that is, similar semantic content of the given document. The BBs are identified by extended compact genetic algorithm (eCGA). In this section, first we give a brief review of eCGA, and then describe our approach to use eCGA on our problem.

Table 2: Reordered DSM matrix.

-	dog	cat	rat	potato
dog	x	1	1	0
cat	1	x	1	0
rat	1	1	x	0
potato	0	0	0	x

6.1 Extended Compact Genetic Algorithm

The extended Compact Genetic Algorithm (eCGA) is probabilistic algorithm which is originated to learn linkage through correcting good probability distributions (7). The measure of desired distribution is quantified based on minimum description length (MDL) model. A class of probability model, marginal product models (MPMs), is used in eCGA for a probability distribution.

The key concept of MDL model is as follows. Simpler distributions are better than the more intricate one, and everything is equal. The MDL restriction penalizes both inaccurate and complex models, therefore conducting to an optimal probability distribution. MPMs are comprise of a product of marginal distributions on a partition of the genes. MPMs also make easier a direct linkage map with each partition separating tightly linked genes.

The eCGA has consecutive sequences as follows.

1. Generate a random population of size N .
2. Undergo tournament selection at rate S .
3. Model the population using a greedy MPM search.
4. If the model has converged, stop.
5. Generate a new population using the given model.
6. Return to step 2.

Two things need further explanation: (1) the determination of MPM using MDL, and (2) the generation of a new population based on MPM. A constrained optimization problem determines MPM in every generation. The determination of MPM in each generation can be represented as minimizing the sum of the model complexity (C_m) and the cost of using single model compare with complex one (C_p). Model complexity C_m and cost of using single model C_p can be represented as follows.

$$C_m = \log_2(n) \sum_{i=1}^m (2^{k_i} - 1) \quad (9)$$

$$C_p = \sum_{i=1}^m \sum_{j=1}^{2^{k_i}} N_{ij} \log_2 \left(\frac{n}{N_{ij}} \right) \quad (10)$$

where n is number of populations, m in the equations represent the number of BBs, k_i is the length of BB $i \in [1, m]$, and N_{ij} is the number of chromosomes in the current population possessing bit-sequence $j \in [1, 2^{k_i}]$ population size.

The greedy MPM search used in eCGA begin with a model that all the variables are independent. Eventually, independent variables merge into subsets till the MDL model no longer progresses. The methodology of generating a new population is as follows, a new population of $n(1 - p_c)$ where p_c is the crossover probability are the best individuals in the current population. The others, $n \cdot p_c$, are filled with randomly choosing subsets from the current individuals based on the probabilities of the subsets.

One of the most important parameters of eCGA is the population size. The analytical population sizing method is developed (16). This analysis estimates the population size which enables to gain the solution as following equation.

$$n \propto 2^k \left(\frac{\sigma_{BB}^2}{d^2} \right) m \log m, \quad (11)$$

where m is number of building blocks, which has size k , failure rate is $\alpha = 1/m$ and $\frac{\sigma_{BB}^2}{d^2}$ is the noise-to-signal ratio (5).

6.2 Semantic BBs Identification by eCGA

Semantic BBs are identified by eCGA method. eCGA is mainly used because its ability to minimize complexity among BBs. eCGA requires vectors as an input. Suppose a given document consisting of paragraphs. Let D be the document which is a set of paragraphs, and denote each paragraph by P_i , that is, $D = \{P_1, P_2, \dots, P_n\}$. Each paragraph P_i is converted into a vector V_i for using eCGA as follows.

Let W be a set of semantic groups obtained from the document and other words (that is, not clustered words). Paragraph vector V_i is defined by

$$V_i = (P_i(w_1), P_i(w_2), \dots, P_i(w_n)) \quad (12)$$

where $w_k \in W$, $1 \leq k \leq n$ and

$$P_i(w_k) = \begin{cases} 0 & \text{if } w_k \text{ is not contained in } P_i \\ 1 & \text{if } w_k \text{ is contained in } P_i \end{cases} \quad (13)$$

This creates a population form from each paragraph. Each binary variable represents the existence of semantic group set in the paragraph. The final step is using eCGA to obtain context-aware semantic building blocks. The building blocks are identified by the MPM. MPM is provided by the model-building process of eCGA.

7 Experiments

For examining our approach, we performed experiments using two documents: 1) a sample document consists of six paragraphs presented in Section 3, and 2) a news article set obtained from Google News on 29th January, 2007. In this section we report the experimental results from each document and show how our approach identify BBs from the documents.

7.1 Experiment 1

As a preliminary experiment, we examined our approach to a small sized sample document, which was shown in Section 3. The sample document consists of six paragraphs.

Table 3: Experiment 1: Semantic groups

Index word	group
describe	[describe write]
meaning	[possible meaning]
one	[one meaning]
solution	[solution meaning words]
words	[words way]
closer	[closer possible]

After the first step of our approach (stop words elimination and stemming words), DSM clustering method identified six semantic groups (see Table 7.1). Each row indicates an index word (left column) of a word group (right column). Each pair (or triplet) has common hypernyms. For example, all of the words “possible”, “meaning” and “one” have a hypernym “cognition” in common.

After clustering the semantic groups provided by DSM, we have a set of 36 semantic groups. Our approach identified two BBs having two words [one paragraph] [describe repeating]. The example paragraphs uses either “describe” or “write”. Only one paragraph uses both, but the semantic layer brought them together. This shows clustering semantic BBs raises efficiency and quality of BBs.

7.2 Experiment 2

To validate our approach, we performed an experiment using a larger document. The document consists of around hundred news articles about *a suicide bombing in Israel* obtained from Google News on Jan 29, 2007.

The stop words elimination reduced the size of word set to half approximately (from 1224 words to 643 words). Then, DSM clustering identified various sizes of semantic groups. Figure 1 shows the number of identified semantic BBs and the sizes of groups (i.e., number of words contained in the group). DSM clustering identified 168 semantic groups. The average size of semantic groups was approximately 5.28.

After DSM clustering step, we have a set of 283 semantic groups identified from the 608 paragraphs available. Table 7.2 shows number of identified BBs of each size. The average size of BBs was approximately 1.74.

Table 4: # of BBs and its size

Size of BBs	1	2	3	4
# of BBs	69	69	24	1

Table 5 shows the top ranked BBs for each size.

Each BBs in Table5 actually represents semantic context in document. In other words, we can have intuitive understanding of the article only by these BBs without reading through. These BBs are well suited for HBGA operation.

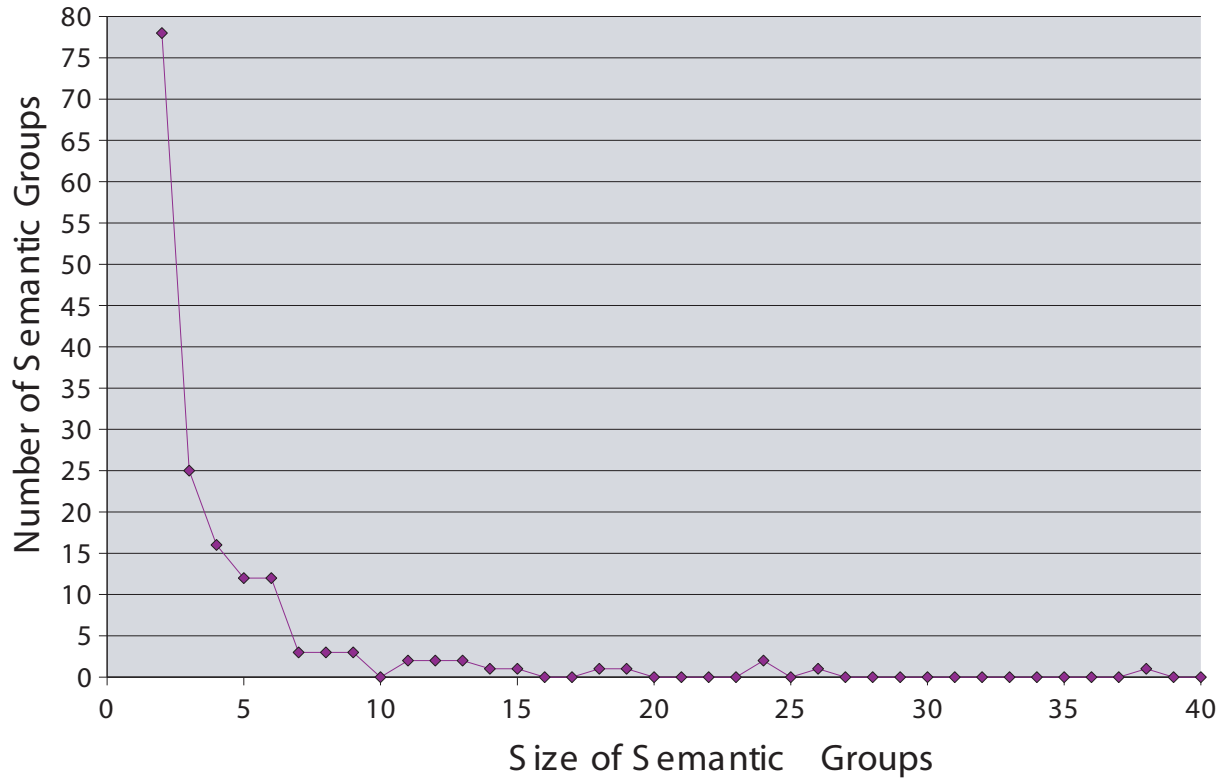


Figure 1: Distribution of Semantic Group Size

Table 5: Examples of identified BBs

Size	BBs
4	[suicide israel april restaurant]
3	[agree occupation march], [gaza strip city], [violence undermine stage], [west bank military], [resort sea tip]
2	[brother thursday], [jihad joint], [iran february], [crackdown fire], [lead ramadan], [exile iranian]

8 Conclusions

In this paper, we proposed an approach for identifying building blocks from text documents. Our approach used DSM clustering methods for finding semantic groups, and eCGA for identifying context-aware semantic building blocks. We report some experimental results that demonstrate how our approach effectively identifies the building blocks from text documents. This work is the first step towards designing a competent HBGA. As the future work, using this work, we will design a competent HBGA for obtaining innovative solutions.

References

- [1] L. D. Baker and A. K. McCallum. Distributional clustering of words for text classification. In W. B. Croft, A. Moffat, C. J. van Rijsbergen, R. Wilkinson, and J. Zobel, editors, *Proceedings of SIGIR-98, 21st ACM International Conference on Research and Development in Information Retrieval*, pages 96–103, Melbourne, AU, 1998. ACM Press, New York, US.
- [2] Dhillon, I., Mallela, S., and Kumar, R. A divisive information-theoretic feature clustering algorithm for text classification. *Journal of Machine Learning Research*, 3:1265–1287, 2003.
- [3] C. Fernandez. Integration analysis of product architecture to support effective team co-location. Master’s thesis, Massachusetts Institute of Technology, Boston, MA, 1998.
- [4] D. Goldberg. *Design of innovation: Lessons from and for competent genetic algorithms*. Kluwer Academic Publishers, Boston, MA, 2002.
- [5] D. E. Goldberg, K. Deb, and J. H. Clark. Genetic algorithms, noise, and the sizing of populations. *Complex Systems*, 6:333–362, 1992. (Also IlliGAL Report No. 91010).
- [6] Goldberg, D.E., Welge, M., and Llorá, X. Discus: Distributed innovation and scalable collaboration in uncertain settings. IlliGAL Report No. 2003017, University of Illinois at Urbana-Champaign, Urbana, IL, June 2003.
- [7] Harik, G.R., Lobo, F.G., and Sastry, K. Linkage learning via probabilistic modeling in the ecga. IlliGAL Report No. 99010, University of Illinois at Urbana-Champaign, Urbana, IL, 1999.
- [8] C. Johansson. Searching for features using a genetic algorithm.
- [9] A. Kosorukoff. Human based genetic algorithm. IlliGAL Report No. 2001004, University of Illinois at Urbana-Champaign, Urbana, IL, March 2001.
- [10] W. B. Langdon. Natural language text classification and filtering with trigrams and evolutionary NN classifiers. In D. Whitley, editor, *Late Breaking Papers at the 2000 Genetic and Evolutionary Computation Conference*, pages 210–217, Las Vegas, Nevada, USA, 8 2000.
- [11] Miller, G., Beckwith, R., Fellbaum, C., Gross, D., and Miller, K. Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography*, 3(4):235–244, 1990.
- [12] K. Mock. Intelligent information filtering via hybrid techniques: Hill climbing, 1996.
- [13] R. Myers and E. R. Hancock. Genetic algorithms for ambiguous labelling problems. In *Energy Minimization Methods in Computer Vision and Pattern Recognition*, pages 345–360, 1997.
- [14] M. Porter. An algorithm for suffix stripping. *Program*, 14(3):130–137, 1980.
- [15] J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978.

- [16] K. Sastry and D. E. Goldberg. Designing competent mutation operators via probabilistic model building of neighborhoods. *Proceedings of the Genetic and Evolutionary Computation Conference*, 2:114–125, 2004. Also IlliGAL Report No. 2004006.
- [17] D. Sharman, A. Yassine, and P. Carlile. Characterizing modular architectures. ASME 14th International Conference, pages DTM-34024, Sept. 2002.
- [18] N. Slonim and N. Tishby. The power of word clusters for text classification. In *23rd European Colloquium on Information Retrieval Research*, 2001.
- [19] D. V. Steward. The design structure system: A method for managing the design of complex systems. *IEEE Transactions on Engineering Management*, 28:77–74, 1981.
- [20] S. M. Weiss, N. Indurkha, T. Zhang, and F. J. Damerau. *Text Mining: Predictive methods for analyzing unstructured information*. Springer, 2005.
- [21] A. Yassine, D. R. Falkenburg, and K. Chelst. Engineering design management: An informatoin structure approach. *International Journal of production research*, 37(13):2957–2975, 1999.