# A Methodology for Feature Selection in Named Entity Recognition

Fredrick Edward Kitoogo*,
Department of Computer Science,
Faculty of Computing and IT,
Makerere University,

Venansius Baryamureeba,
Department of Computer Science,
Faculty of Computing and IT,
MakerereUniversity.

In this paper a method for feature selection in named entity recognition is proposed. Unlike traditional named entity recognition approaches which mainly consider accuracy improvement as the sole objective, the innovation here is manifested in the use of a multi-objective genetic algorithm which is employed for feature selection basing on various aspects including error rate reduction and time taken for evaluation, and also demonstrating the use of Pareto optimization. The proposed method is evaluated in the context of named entity recognition, using three different data sets and a K-nearest Neighbour machine learning algorithm. Comprehensive experiments demonstrate the feasibility of the methodology.

Categories and Subject Descriptors: I.5.2 [**Pattern Recognition**]: Design Methodology—Feature evaluation and selection; I.2.7 [**Artificial Intelligence**]: Natural Language processing—language parsing and understanding

General Terms: Computer Science, Language Processing
Additional Key Words and Phrases: feature selection, multi-objective genetic algorithm, named entity recognition.

**IJCIR Reference Format:**
Kitoogo, F.E. and Baryamureeba, V. 2007.A Methodology for Feature Selection in Named Entity Recognition. International Journal of Computing and ICT Research, Vol.1, No. 1, pp. 18-26.
http://www.ijcir.org/volume1-number1/article3.pdf

## 1. INTRODUCTION

The Machine Learning approaches to the named entity recognition (NER) problem follow three major steps namely; (i) feature engineering, where identification of lexical and phrasal characteristics in text which expresses references to named entities (NEs) is done, (ii) algorithm selection, when the decision of

* Author's Address: Fredrick Edward Kitoogo*, Department of Computer Science, Faculty of Computing and IT, Makerere University, P.O. Box 7062, Kampala, Uganda, *fkitoogo@cit.mak.ac.ug, www.cit.ac.ug*
Venansius Baryamureeba, Department of Computer Science, Faculty of Computing and IT, Makerere University, P.O. Box 7062, Kampala, Uganda, *barya@cit.mak.ac.ug, www.cit.ac.ug*


International Journal of Computing and ICT Research, ISSN 1818-1139, Vol.1, No.1, pp. 18 - 26, June 2007

which machine learning algorithm/algorithms to use for learning is made and (iii) classification, when the actual learning of the feature list to detect and classify the named entity phrases is done.
NEs are theoretically identified and classified by using features (various abstract entities that combine to specify underlying phonological, morphological, semantic, and syntactic properties of linguistic forms and that act as the targets of linguistic rules and operations). Two kinds of features that have been defined by McDonald [1996] are internal and external features; internal features are the ones provided from within the sequence of words that constitute the entity, in contrast, external features are those that can be obtained by the context in which entities appear.

The choice of the best discriminative features to represent NEs affects many aspects of the NER problem such as accuracy, learning time, and the optimal training data set size. In many NER applications, it is not unusual to find problems involving hundreds of features. However, it has been observed that beyond a certain point, the inclusion of additional features leads to a worse rather than better performance [Oliveira et al. 2003].

Feature engineering refers to the task of identifying and selecting an effective subset of features to represent entities from a larger set of often mutually redundant or even irrelevant features. It encompasses feature design, feature selection, feature induction, and feature impact optimization [Rinnger 2005]. Feature selection; a sub-task of feature engineering is not a trivial problem since there may be (i) redundancy, where certain features are correlated so that it is not necessary to include all of them in modeling and (ii) interdependence, where two or more features between them convey important information that is obscure if any of them is included on its own.
Many real-world problems like feature selection for named entity recognition involve the optimization of multiple objectives, such as number of features and accuracy. The tendency is that the different objectives to be optimized represent conflicting goals (such as improving the quality of a product and reducing its cost),in multi-objective optimization the optimization of each objective corresponds to an optimal solution. Therefore, in multi-objective optimization one usually wants
to discover several optimal solutions, taking all objectives into account, without assigning greater priority to one objective or the other. Most named entity recognition systems as demonstrated in Tjong Kim Sang and De Meulder [2003] tend to consider only one objective of improving accuracy.

There is need for systems which put into consideration other objectives like the cost of the solution on top of improvement of accuracy. The systems should be further able to provide users with different sets of optimal solutions thus giving the end-user the option of being able to choose the solution representing the best trade-off between conflicting objectives a posteriori, after examining a set of high-quality solutions returned by the named entity recognition system. Intuitively, this is better than forcing the user to choose a trade-off between conflicting goals a priori.

This paper proposes the use of a multi-objective genetic algorithm (MOGA) as a means to search for subsets of features (feature selection). The MOGA will generate a feature set of alternative solutions (from a fixed entire feature population) and
use a cross-validation method to indicate the best accuracy/complexity (number of
Features)/cost of using the feature sub-set (in this case only time for classification was used as a cost) trade-off. The classification accuracy will be supplied by a machine learning algorithm.

The remainder of the paper is structured as follows: in Section 2, previous approaches are reviewed, in Section 3, we outline the proposed method and specify the feature population from which feature selection will be done. We describe the search (optimization) procedure of the method, in Section 4 the experiments and results are presented. Finally, Section 5 closes with a conclusion and an outlook for future work.

## 2. PREVIOUS APPROACHES

Many researchers have tackled feature selection in various ways and likewise the performance of the different approaches varies substantially. The more closely related approaches are presented in this section.

### 2.1 Complete Search

Some researchers manually designed features and used all of them without selecting optimal subsets [Carreras et al. 2003; Zhou et al. 2004; Shen et al. 2004]

, while others leave the task of ignoring the useless features to the learning algorithm [Mayfield et al. 2003]. The problem with these approaches, is that they are computationally not feasible in practice.

### 2.2 Randomized Search

Randomized algorithms make use of randomized or probabilistic steps or sampling processes. Several researchers have explored the use of such algorithms for feature selection [Kira et al. 1992; Liu et al. 1996]. Li and McCallum [2004] use conjunctions of the original features; where they use feature induction which aims to create only those conjunctions which significantly improve performance by starting with no features at all and iteratively choosing new features, from which sets are built and correspondingly evaluated at each iteration. Hakenberg et al. [2005] tackle feature engineering using what they refer to as recursive feature elimination (RFE); where they study the impact of gradual exclusion of features. Their model starts with a full model containing all features, they iteratively remove a number of features with the lowest weight, retrain the model, and check the resulting performance. Bender [2003] use count-based feature reduction; where a threshold K is predetermined, and only those features that have been observed in the training data set at least K times are considered for the learning algorithm. Jiang and Zhai [2006] use what they termed generalizability-based feature ranking; here they target selecting features which are more generalizable (perform well in different domains). They use generalizability to mean the amount of contribution a feature can make to the classification accuracy on any domain, and to identify highly generalizable features, they compare their individual contributions to accuracy among different domains using a predefined scoring function.

The problem with the randomized search techniques is that they cannot properly handle the interdependence and correlation problem often associated with feature selection from large search spaces because they do not explore the whole search space at once.

### 2.3 Heuristic Search

Several authors have explored the use of heuristics for feature subset selection, often in conjunction with branch and bound search. Forward selection and backward elimination are the most common sequential branch and bound search algorithms used in feature selection [John et al. 1994]. Most of these approaches assume monotonicity of some measure of classification performance. This ensures that adding features does not worsen the performance. However, many practical scenarios do not satisfy the monotonicity assumption. Moreover, this kind of searchis not designed to handle multiple selection criteria.

Another branch of heuristic approaches employ genetic algorithms which do not require the restrictive monotonicity assumption. They can also deal with the use of multiple selection criteria, e.g. Classification accuracy, feature measurement cost, etc. Due to the ability of genetic algorithms to deal with multi-objective optimization, some authors have explored genetic algorithms for feature selection for handwritten character recognition [Kim et al. 2000].

Feature selection using genetic algorithm is often performed by aggregating different objectives into a single and parameterized objective, which is achieved through a linear combination of the objectives. The main drawback of this approach is that it is very difficult to explore different trade-offs between accuracy and different subsets of selected features. In order to overcome this kind of problem, Emmanouilidis et al. [2000] proposed the use of a multi-criteria genetic algorithm to perform feature selection.

Li et al. [2005] presented a gene selection approach based on a hybrid between genetic algorithms and support vector machines. The major goal of their hybridization was to exploit fully their respective merits (e.g., robustness to the size of solution space and capability of handling a very large dimension of feature genes) for identification of key feature genes (or molecular signatures) for a complex biological phenotype. Jirapech-umpai and Aitken [2006] designed an evolutionary algorithm to identify the near-optimal set of predictive genes that classify micro-array data, for multiple objectives of their problem they used a weighting function to compute the fitness of an individual in the population.

Hong and Cho [2006] noted the problem of conventional feature selection with genetic algorithms in handling huge-scale feature selection. They modified the representation of a chromosome to be suitable for huge-scale feature selection and adopted speciation to enhance the performance of feature selection by obtaining diverse solutions.

## 3. THE PROPOSED METHODOLOGY

A Genetic Algorithm (GA) [Goldberg 1989] is a search algorithm inspired by the principle of natural selection. The basic idea is to evolve a population of individuals, where each individual is a candidate solution to a given problem. Each individual is evaluated by a fitness function, which measures the quality of its corresponding solution. At each generation (iteration) the fittest (the best) individuals of the current population survive and produce offspring resembling them, so that the population gradually contains fitter and fitter individuals i.e., better and better candidate solutions to the underlying problem [Paapa et al. 2002].

The work proposes a methodology that employs a multi-objective genetic algorithm (MOGA) as a means to select subsets of features from a pool of pre-designed features, which contain the most optimal discriminatory information to classify named entities. The inspiration of employing a MOGA for feature selection, was that: (i) GAs are a robust search method, capable of effectively exploring the large search spaces often associated with feature selection problems; (ii) GAs perform a global search [Paapa et al. 2002], so that they tend to cope better with feature correlation and interdependence; and (iii) GAs are suitable for multi-objective problem solving [Morita et al. 2003], where the search algorithm is required to consider a set of optimal solutions at each iteration.

The general purpose multi-objective genetic algorithm is to search for optimal subsets of relevant features that minimize the classification error rate, the classifier evaluation time involved, number of learning examples, number of features, and the cost of learning. This work is however, limited to the minimization of classification error rate and the classifier evaluation time involved.

Basically the proposed methodology as shown in Table 1 works as follows: A MOGA will generate from a feature population, alternative feature set solutions from which the best fitness (error rate/time for evaluation of using the feature subset trade-off) will be determined. The fitness of all the individuals in the population will be ranked and a set of optimal individuals will be passed on for crossing- over and subsequent mutation. Pareto Optimal solutions will be selected for each generation. Consequently if the termination condition of the MOGA is reached the algorithm stops and gives the final solution. The classification error rate and time will be supplied by the evaluation of a classifier built from a learning algorithm using 10 fold cross-validation on the training data set. The fitness function in this problem will be based on the error rate and time taken to evaluate the classifier which will be the objectives to be minimized using a Pareto optimization approach.

### 3.1 Individual Representation

Any candidate subset of selected features is represented as an individual which consists of N genes (equating to the total number of all features in the population). Each of the gene values is either turned on or off (0 or 1) for presence or absence in the candidate selected feature subset.

### 3.2 Fitness Function

The fitness function is for measuring the quality of an individual (a candidate feature subset). There are two quality measures for an individual; (i) the error rate of the classification algorithm and (ii) the cost of using the feature set (in this case only time spent for evaluation was used as a cost). The two quality measures are computed by evaluating the average classifier algorithm error rate using cross- validation over ten folds and the time taken to evaluate a classifier using the specific feature set.

### 3.3 Pareto Optimization

A promising approach for performing feature selection is the multi-objective GAs aiming at producing solutions with Pareto optimization suggested in EvoGrid[1]. The key concept here is dominance; to illustrate the concept, the multi-objective problem is represented mathematically in Eq. (1)

[1] Multi-objective optimization with EvoGrid; http://champiland.homelinux.net/evogrid/doc/multiobjective.html, accessed 15/12/2006.

$$Minimize/Maximize\ F(x) = [f_1(x), f_2(x), f_3(x), ..., f_o(x)] \quad (1)$$

Where; $F(x)$ is a multi-objective function vector, $f_i(x)$ is the $i^{th}$ objective function, $x$ is an input vector, $o$ is the number of objective functions.

A solution is said to be Pareto optimal if it cannot be dominated by any other solution available, i.e. A solution $x_i \in X$ is Pareto optimal *iff* there is no $x_j \in X$ such that $f_p(x_j) \leq f_p(x_i)$ $\forall p \in \{1, 2, 3, ....o\}$

The use of a multi-objective algorithm basing on the concept of dominance can maintain population diversity, which allows for the discovery of a range of feature sets using accuracy/cost (time for evaluation) trade-offs. There are two popular

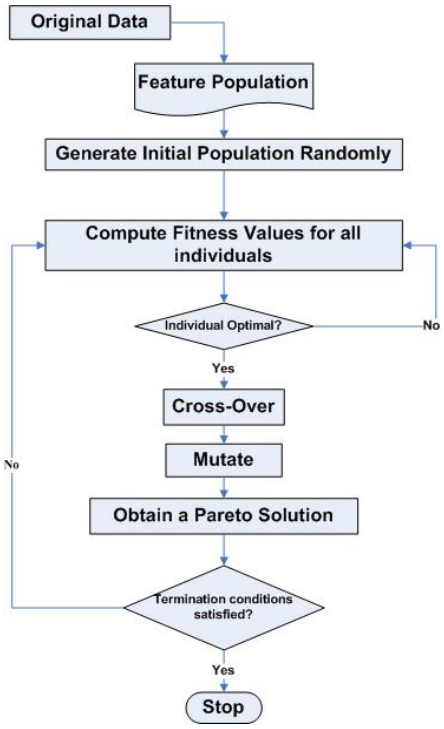


Fig. 1. Flow chart of the proposed methodology

| Datasets used in the Experiments | | | |
|---|---|---|---|
| Datasets | # Instances | # Classes | # Features |
| Breast Cancer | 286 | discrete/2 | 9 |
| Hayes-roth | 132 | discrete/3 | 4 |
| Zoo | 101 | discrete/7 | 16 |

Table I. Details of the datasets used in the experiments

Pareto dominance techniques:-

—Weak Pareto Dominance: A vector $F_m(x)$ weakly Pareto-dominates another vector $F_n(x)$ if none of the $F_n(x)$ coordinates is strictly greater than those in $F_m(x)$ and at least one of the coordinates in $F_m(x)$ is strictly greater than its counterpart $F_n(x)$.

—Strong Pareto Dominance: Here, all components of a dominant vector $F_m(x)$ are strictly greater than their counterpart in $F_n(x)$.

## 4. EXPERIMENTS AND RESULTS

### 4.1 The Data Set

Three real-world datasets (shown in Table I) from the UCI MLR[2] were used for the experiments. Different feature sets for experimentation were built randomly out of the entire feature population of each data set.

### 4.2 The Learning Algorithm

A K-Nearest Neighbour machine learning algorithm (with default parameter settings) was employed for this work.

### 4.3 The Parameter Settings

In our experiments, the MOGA is based on bit individual representation, mutation, crossover and tournament selection based on weak-Pareto optimization. Below are the parameter settings used in the experiments:

- Population size: 100
- Number of Generations: 10
- Probability of crossover (Two-Point): 0.9
- Probability of mutation (Bit Flip): 0.1
- Tournament Size: 8
- Pareto Optimization: Weak

### 4.4 Results

The main results of our experiments are reported in Table II; the Table is divided into three Sub-Tables, each representing a particular data set. For each data set five runs of the MOGA were made; in each run, a comparison of performance (the number of features, error rate, and time spent on evaluation) between the original

| Hayes-Roth Data Set; Instances = 132 | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| Run | Original Features | Feature # | Error Rate | Time (Secs) | | Selected Features | Feature # | Error Rate | Time (Secs) |
| Run 1 | 1111 | 4 | 0.3044 | 0.0160 | | 1111 | 4 | 0.3044 | 0.0149 |
| Run 2 | 1111 | 4 | 0.3044 | 0.0620 | | 1111 | 4 | 0.3044 | 0.0149 |
| Run 3 | 1111 | 4 | 0.3044 | 0.0630 | | 1111 | 4 | 0.3044 | 0.0149 |
| Run 4 | 1111 | 4 | 0.3044 | 0.0620 | | 1111 | 4 | 0.3044 | 0.0149 |
| Run 5 | 1111 | 4 | 0.3044 | 0.0469 | | 1111 | 4 | 0.3044 | 0.0149 |
| Average | | | 0.3044 | 0.0500 | | | | 0.3044 | 0.0149 |
| Std. Dev. | | | 0.0000 | 0.0201 | | | | 0.0000 | 0.0000 |
| | | | | | | | | | |
| Breast Cancer Data Set; Instances = 286 | | | | | | | | | |
| Run | Original Features | Feature # | Error Rate | Time (Secs) | | Selected Features | Feature # | Error Rate | Time (Secs) |
| Run 1 | 111111111 | 9 | 0.3005 | 0.1099 | | 011001110 | 5 | 0.2793 | 0.0780 |
| Run 2 | 111111111 | 9 | 0.3005 | 0.1089 | | 000010100 | 2 | 0.2311 | 0.0779 |
| Run 3 | 111111111 | 9 | 0.3005 | 0.1089 | | 010110110 | 5 | 0.2797 | 0.0929 |
| Run 4 | 111111111 | 9 | 0.3005 | 0.1870 | | 010011001 | 4 | 0.2369 | 0.0779 |
| Run 5 | 111111111 | 9 | 0.3005 | 0.1100 | | 000010100 | 2 | 0.2311 | 0.0779 |
| Average | | | 0.3005 | 0.1249 | | | | 0.2516 | 0.0809 |
| Std. Dev. | | | 0.0000 | 0.0347 | | | | 0.0256 | 0.0067 |
| Zoo Data Set; Instances = 101 | | | | | | | | | |
| Run | Original Features | Features # | Error Rate | Time (Secs) | | Selected Features | Features # | Error Rate | Time (Secs) |

[2] UCI MachineLearning Repository; http://www.cs.uci.edu/mlearn/MLRepository.html, accessed 04/05/2006

| | | | | | | | | |
|---|---|---|---|---|---|---|---|---|
| Run 1 | 1111111111111111 | 16 | 0.0291 | 0.0160 | 1111010011001010 | 9 | 0.0091 | 0.0001 |
| Run 2 | 1111111111111111 | 16 | 0.0291 | 0.0469 | 1111110100111101 | 12 | 0.0100 | 0.0149 |
| Run 3 | 1111111111111111 | 16 | 0.0291 | 0.0469 | 1010110010011010 | 8 | 0.0190 | 0.0001 |
| Run 4 | 1111111111111111 | 16 | 0.0291 | 0.0469 | 1111110011001111 | 12 | 0.0091 | 0.0149 |
| Run 5 | 1111111111111111 | 16 | 0.0291 | 0.0320 | 1111110011001111 | 12 | 0.0091 | 0.0149 |
| Average | | | 0.0291 | 0.0377 | | | 0.0113 | 0.0090 |
| Std. | | | 0.0000 | 0.0138 | | | 0.0043 | 0.0081 |

Table II. Performance of the K-Nearest Neighbour using a MOGA on the three different data sets

feature set and the one generated by the MOGA is made. The Left Hand Side of the Tables shows the performance of the original feature set while the Right Hand Side shows that using the feature set selected by the MOGA. In the columns ”Original Features” and ”Optimal Features” which are a bit representation of the used features, a ”1” indicates presence of a feature in order of position, while ”0” denotes absence of that position feature. The aim of the MOGA is to minimize both the error rate and time, implying that a value depicts better performance than another (comparable one) when the former value is less than the latter value.

Clearly, as shown in Table II the performance (error rate and time for evaluation) for all runs in two data sets [Breast Cancer and Zoo] is better when using the MOGA selected feature sets than with all features. In general the average performance for the MOGA selected feature sets in the two data sets is significantly (a value is significantly better than another when the corresponding upper and lower bounds using the standard deviation do not overlap) better than that with all features; this finding supports the fact of inclusion of only optimal features in a set, which precipitates the assertion of many researchers that interdependence and correlation in target features affect feature selection task. The performance on the Hayes-Roth Data Set is actually identical, except the time to evaluate which is marginally significantly divergent.

| Cancer Data Set - Run 1 | | | | Zoo Data Set - Run 4 | | | |
|---|---|---|---|---|---|---|---|
| Selected Feature Set | Features numbers | Error Rate | Time (Secs) | Selected Feature Set | Features numbers | Error Rate | Time (Secs) |
| 001101000 | 3 | 0.2974 | 0.0780 | 0110110001001100 | 7 | 0.0291 | 0.0001 |
| 011001110 | 5 | 0.2793 | 0.0929 | 0101110000011100 | 7 | 0.0291 | 0.0001 |
| 000101000 | 2 | 0.2898 | 0.0779 | 1111110011001111 | 12 | 0.0091 | 0.0149 |

Table III. Pareto Optimal Solutions for the a K-nearest Neighbour machine learning algorithm on the Breast Cancer and Zoo Data Sets

| All Data Sets | | | | |
|---|---|---|---|---|
| Date Set | Original Feature Set Size | Original Average Error Rate [C3] | MOGA Average Error Rate [C4] | % age Error Rate Improvement C4 − C3 [(\|————\|) □ 100] |
| Hayes-Roth | 4 | 0.2974 | 0.0780 | 0 |
| Breast Cancer | 9 | 0.2793 | 0.0929 | 16.26 |
| Zoo | 16 | 0.2898 | 0.0779 | 61.30 |

Table IV. Comparison of Percentage Improvement in Average Error Rate of all three Data Sets

Table III demonstrates the concept of *Pareto Dominance*, which explains the difficulty in determining trade-offs between several optimization objectives used in other approaches like the simple weighted approach; in the Breast Cancer data set it is seen that the second result has a better error rate than the first result, however the time component results are the opposite. A similar situation is shown in the Zoo Data Set where the error rate for the third result is better than that of the first and second result but the time performance is the opposite. In both situations the feature sets performances are better than those of all the other possible feature sets (non-dominated) but a better solution between themselves is not easily identifiable

a priori. The situation here can only be determined a posteriori by an end-user for a perceived better option (using a choice of their best error rate/ Time trade-off).

Table IV shows that as the number of features increases in a data set, MOGA performs (error rate) better, with this observation it is possible to conclude that employment of is this technique is best suited for data sets with a large number of original features.

## 5. CONCLUSION AND FUTURE WORK

We have introduced a methodology to perform feature selection for the classification task in named entity recognition based on a multi-objective genetic algorithm. We have experimented this approach with the application of a weak Pareto-tournament selection genetic algorithm and a k-Nearest Neighbour machine learning algorithm and demonstrated its efficacy on three real world data sets. We have shown that the multi-objective genetic algorithm is well suited for feature selection and that it has a benefit of yielding different solution options based on error rate/cost trade-offs, leaving end-users with an option of deciding from alternative solutions.

It has also been discovered that multi-objective genetic algorithms are justifiably more suitable for feature selection where the number of features is large enough to make other methods computationally more expensive.

Future work should examine other approaches to Pareto optimization, and more pragmatic approaches to determining optimal population and generation sizes for experiments. Studies about specific feature impact optimization and interdependence need to be carried out. Deeper comparison of weighted approaches to multi- objective genetic algorithms should be considered. The multi-objective genetic algorithm methodology to feature selection for named entity recognition offers more potential for considering many more objectives in the named entity recognition task other than accuracy improvement only.

## REFERENCES


BENDER, O., OCH, F. J. AND NEY, H. 2003. Maximum Entropy Models for Named Entity Recognition. In *Proceedings of CoNLL-2003 on language independent natural language processing*, 148 - 151.

Carreras, X., MARQUEZ, L. AND PADRO, L. 2003. A simple named entity extractor using AdaBoost. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003*, 4: 152-155.

EMMANOUILIDIS, C., HUNTER, A., AND MACLNTYRE, J. 2000. A multi-objective evolutionary setting for feature selection and a commonality-based crossover operator. In *Proceedings of Congress on Evolutionary Computation*, 1: 309 - 316.

GOLDBERG, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley Publishing Company

HAKENBERG, J., BICKEL, S., PLAKE, C., BREFELD, U., ZAHN, H., FAULSTICH, L., LESER, U., AND Scheffer, T. 2005. Systematic feature evaluation for gene name recognition. *BMC Bioinformatics*, 6(l):S9.

HONG, J. H., AND CHO, S. B. 2006. Efficient huge-scale feature selection with speciated genetic algorithm. *Pattern Recognition Letters*, 27: 143 - 150.

JIANG, J. AND ZHAI, C. X. 2006. Exploiting Domain Structure for Named Entity Recognition. Urbana, 51: 61801.

JIRAPECH-UMPAI, T. AND AITKEN, S. 2006. Feature selection and classification for microarray data analysis: Evolutionary methods for identifying predictive genes. *BMC Bioinformatics*, 6:148.

JOHN, G., KOHAVI, R., AND PEGER, K. 1994. Irrelevant features and the subset selection problems. In *Proceedings of the 11th International Conference on Machine Learning*, 121 - 129.

KIM, G., AND KIM, S. 2000. Feature selection using genetic algorithms for handwritten character recognition. In *Proceedings of the 7th International Workshop on Frontiers of Handwriting Recognition (IWFHR)*, 103 - 112.

KIRA, L., AND RENDELL, L. 1992. A practical approach to feature selection. In *Proceedings of the 9th International Conference on Machine Learning*, 249-256

LI, W. AND MCCALLUM, A. 2004. Rapid development of Hindi named entity recognition using conditional random fields and feature induction. In *ACM Transactions on Asian Language Information Processing (TALIP)*, 2: 290-294.

LI, L., JIANG, W., LI, X., MOSER, K. L., GUO, Z., DU, L., WANG, Q., TOPOL, E. J. and RAO, S. 2005. A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset. Genomics, 85: 16-23.

LIU, H., AND SETIONO, R. 1996. A probabilistic approach to feature selection a filter approach. In *Proceedings of the 13th International Conference on Machine Learning,* 319 - 327.

MAYFIELD, J., MCNAMEE, P. and PIATKO, C. 2003. Named entity recognition using hundreds of thousands of features. In *Proceedings of CoNLL-2003*, 184 - 187.

MCDONALD, D. 1996. Internal and external evidence in the identification and semantic categorization of proper names. *Corpus Processing for Lexical Acquisition*, 21 - 39.

MORITA, M., SABOURIN, R., BORTOLOZZI, F., SUEN, C. Y. AND DE TECHNOLOGIE SUPERIEUE, E. 2003. Unsupervised feature selection using multi-objective genetic algorithms for handwritten word recognition. In *Proceedings of the Seventh International Conference on Document Analysis and Recognition*, 666-670.

OLIVEIRA, L. S., SABOURIN, R., BORTOLOZZI, F., and SUEN, C. Y. 2003. A Methodology for Feature Selection Using Multi-Objective Genetic Algorithms for Handwritten Digit String Recognition. *International Journal of Pattern Recognition and Artificial Intelligence*, 17: 903 - 929

PAPPA, G. L., FREITAS, A. A. AND KAESTNER, C. A. A. 2002. Attribute Selection with a Multi-objective
Genetic Algorithm. In *XVI Brazilian Symposium on Artificial Intelligence*. 2057: 280 - 290.

RINNGER, E. 2005. The *ACL 2005 Workshop on Feature Engineering for Machine Learning in Natural Language*.

SHEN, D., ZHANG, J., SU, J., ZHOU, G., TAN, C. 2004. Multi-Criteria-based Active Learning for Named Entity Recognition.

TJONG KIM SANG, E.F. AND DE MEULDER, F. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, 142 - 147

ZHOU, G.D., SHEN, D., ZHANG, J., SU, J. AND TAN, S.H. 2004. Recognition of Protein/Gene Names from Text using an Ensemble of Classifiers. *BMC Bioinformatics*.