Evolutionary Optimization for NLP

Asif Ekbal

Department of
Computer Science and Engineering
IIT Patna, India-800 013
Email: asif@iitp.ac.in
asif.ekbal@gmail.com

Outline

- Introduction to Optimization Techniques
    - Single vs Multiobjective Optimization

- Evolutionary Algorithms to NLP
- Introduction to Genetic Algorithm
- SOO based Classifier Ensemble
    - Weighted vote based classifier ensemble

- NE Extraction in biomedicine
    - Features
    - Results

Outline

- Cross-corpus compatibilities in biomedicine
- Ensemble using Multiobjective Optimization
- MOO based techniques for anaphora resolution
    - Feature selection
    - Learning algorithm selection
    - Parameter optimization
    - Architecture selection
    - Experiments and results

What is Optimization?

Optimization is the process of minimizing or maximizing a function subject to several constraints on its variables

Optimization

- In our every day life, we make decisions consciously or unconsciously
- Decision can be
  - very simple such as selecting the colour of dress or deciding the menu for lunch (*very easy to take*)
  - may be as difficult as those involved in designing a missile or in selecting a career (*might take several years due to the level of complexity involved in it*)
- Main goal of most kinds of decision-making is to optimize

  one or more criteria in order to achieve the desired result

In other words, problems related to optimization abound in real life

Single vs. Multi-objective

Single Objective Optimization:
When an optimization problem involves only one objective function, the task of finding the optimal solution is called single-objective optimization

Example: Find out a **CAR** for me with Minimum cost

Multi-objective Optimization: When an optimization problem involves more than one objective function, the task of finding one or more optimal solutions is known as multi-objective optimization

Example**:** Find out a **CAR** with minimum cost and maximum comfort

Multiobjective Optimization: Mathematical Definition

The multiobjective optimization can be formally stated as:

Find the vector of decision variables

$$\mathbf{x}=[x_1, x_2\ldots\ldots x_n]^T$$

which will satisfy the m inequality constraints:

$$g_i(\mathbf{x}) >= 0,\ i=1,2,\ldots.m,$$

And the p equality constraints

$$h_i(\mathbf{x})=0\ ,\ i=1,2,\ldots.p.$$

And simultaneously optimizes M objective functions

$$f_1(\mathbf{x}), f_2(\mathbf{x})\ldots.\ f_M(\mathbf{x}).$$

Pareto Optimum: Definition

- A candidate is *Pareto optimal* iff:
  - It is *at least as good* as all other candidates for all objectives, and
  - It is *better than all other candidates* for at least one objective
- We would say that this candidate *dominates* all other candidates

Dominance: Definition

Given the vector of objective functions $\vec{f}(\vec{x}) = (f_1(\vec{x}), \ldots, f_k(\vec{x}))$

we say that candidate $\vec{x}_1$ dominates $\vec{x}_2$, (i.e. $\vec{x}_1 \preceq \vec{x}_2$) if:

$$f_i(\vec{x}_1) \leq f_i(\vec{x}_2) \quad \forall i \in \{1, \ldots, k\}$$

$$\mathit{and}$$

$$\exists i \in \{1, \ldots, k\} : f_i(\vec{x}_1) < f_i(\vec{x}_2)$$

(assuming we are trying to minimize the objective functions)

(Coello Coello 2002)

Pareto Optimal Set

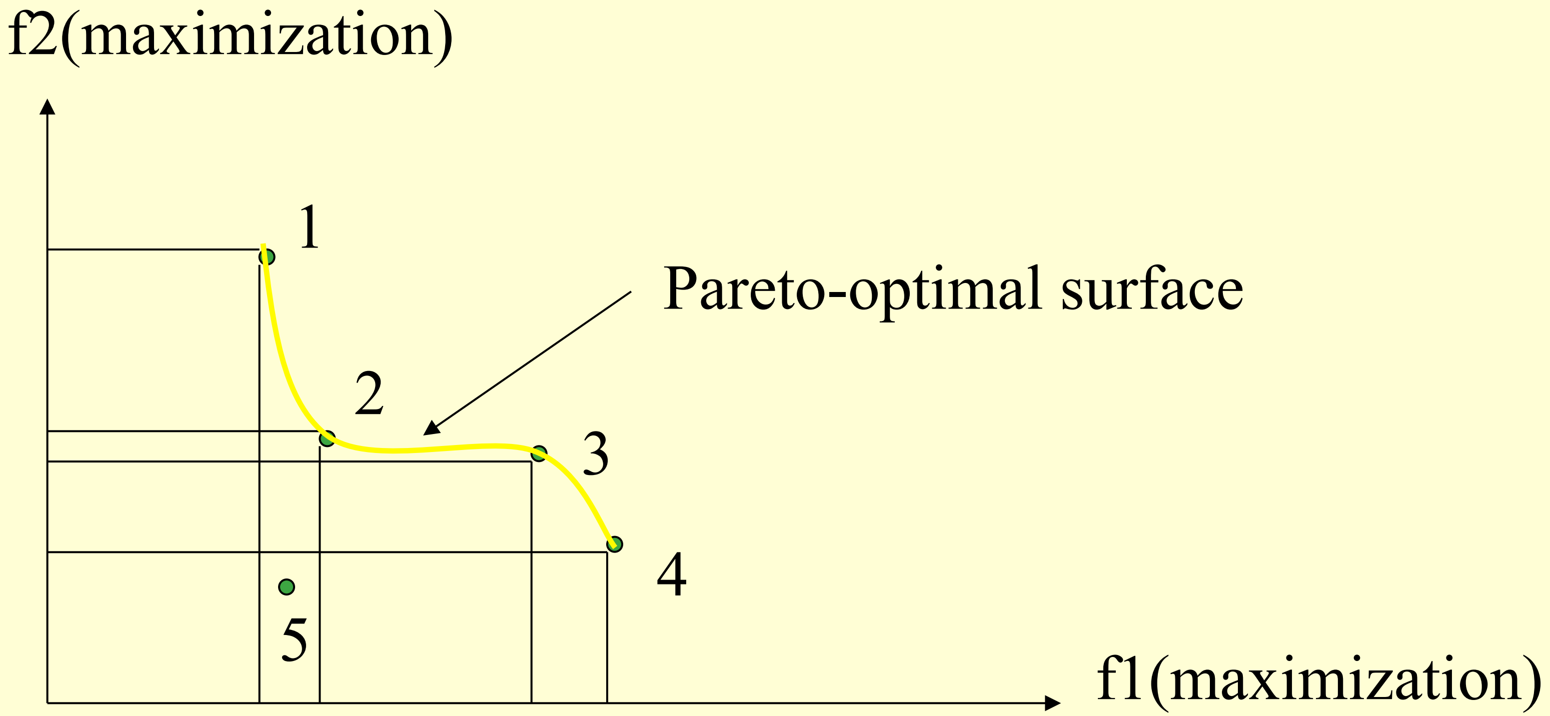
The Pareto optimal set *P* contains all candidates that are non-dominated. That is:

$$P := \left\{ x \in F \,\middle|\, \left[\neg \exists x' \in F \right] \ni \left(\vec{f}(x') \preceq \vec{f}(x) \right) \right\}$$

where *F* is the set of feasible candidate solutions

(Coello Coello 2002)

Example of Dominance and Pareto-Optimality



- Here solutions1, 2, 3 and 4 are non-dominating to each other.
- 5 is dominated by 2, 3 and 4, not by 1.

Promising Solutions

- Meta-heuristics
  - Evolutionary algorithms
  - Simulated annealing

- Have shown promise in solving complex *single* as well as *multiobjective optimization* problems in a wide variety of domains

Evolutionary Algorithms in NLP

- Good Review (L. Araujo, 2007)
- Natural language tagging- Alba, G. Luque, and L. Araujo (2006)
- Grammar Induction-T. C. Smith and I. H. Witten (1995)
- Phrase-structure-rule of natural language-W. Wang and Y. Zhang (2007)
- Information retrieval-R. M. Losee (2000)
- Morphology -D. Kazakov (1997)
- Dialogue systems-D. Kazakov (1998)
- Grammar inference -M. M. Lankhors (1994)
- Memory-based language processing (A. Kool, W. Daelemans, and J. Zavrel., 2000)

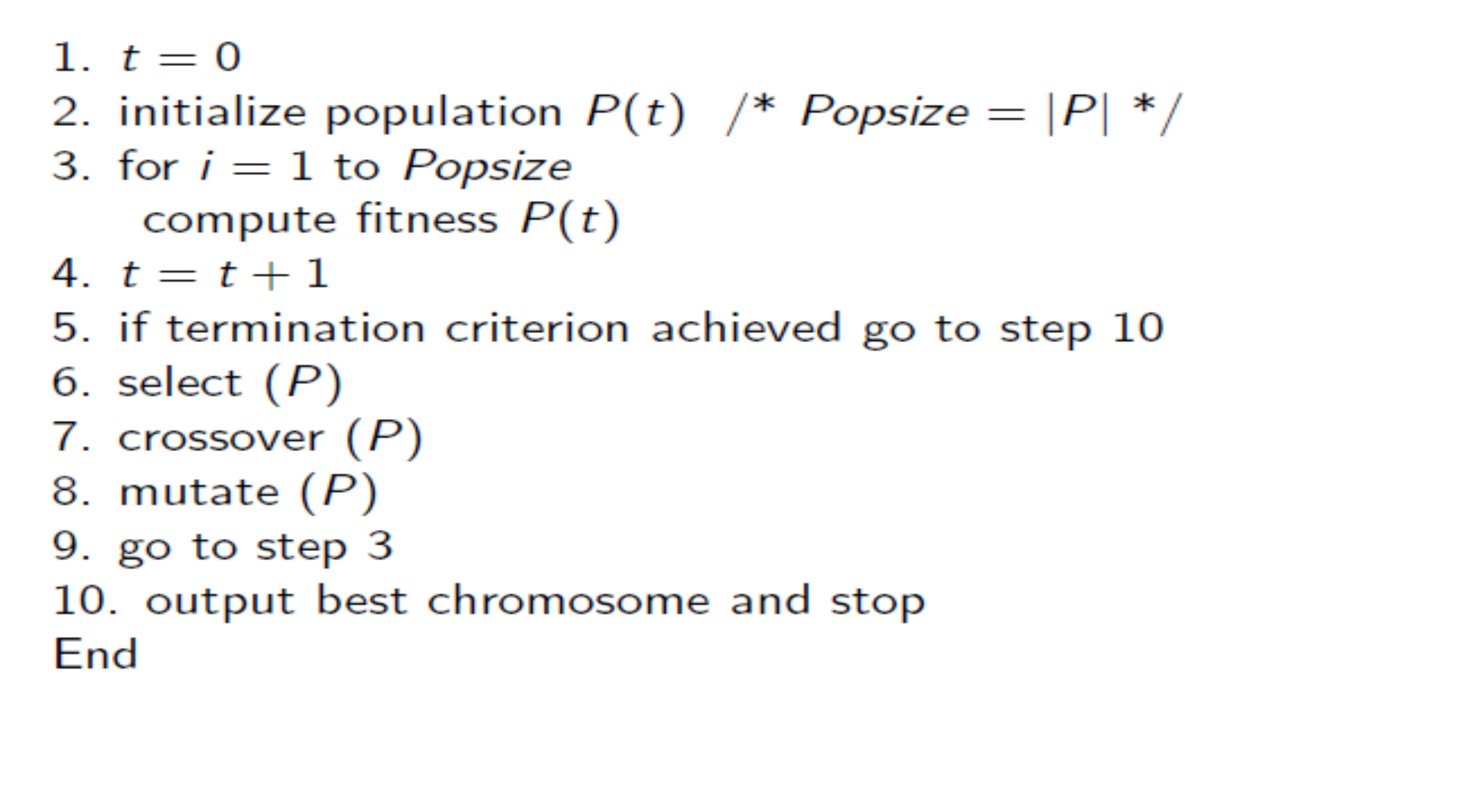
Evolutionary Algorithms in NLP

- Anaphora resolution:Veronique Hoste (2005), Ekbal et al. (2011), Saha et al. (2012)
- Part-of-Speech tagging: Araujo L (2002)
- Parsing: Araujo L (2004)
- Document clustering: Casillas A et al. (2003)
- Summarization: Andersson L (2004)
- Machine Translation : Jun Suzuki (2012)
- NER: Ekbal and Saha (2010; 2011; 2012 etc.)

Genetic Algorithm: Similarity with Nature

Genetic Algorithms	←→	Nature
A solution (phenotype)		Individual
Representation of a solution (*genotype*)		Chromosome
Components of the solution		Genes
Set of solutions		Population
Survival of the fittest (*Selection*)		Darwins theory
Search operators		Crossover and mutation
Iterative procedure		Generations

Basic Steps of Genetic Algorithm



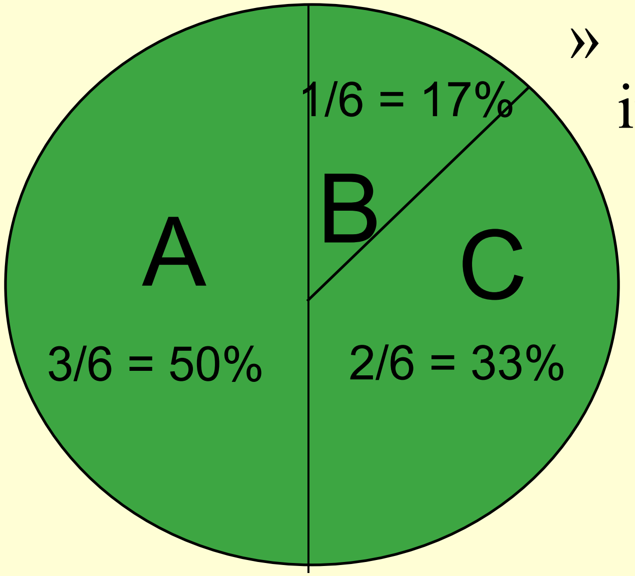
```
1. $t = 0$
2. initialize population $P(t)$  /* $Popsize = |P|$ */
3. for $i = 1$ to $Popsize$
     compute fitness $P(t)$
4. $t = t + 1$
5. if termination criterion achieved go to step 10
6. select $(P)$
7. crossover $(P)$
8. mutate $(P)$
9. go to step 3
10. output best chromosome and stop
End
```

Example population

No.	Chromosome	Fitness
1	1010011010	1
2	1111100001	2
3	1011001100	3
4	1010000000	1
5	0000010000	3
6	1001011111	5
7	0101010101	1
8	1011100111	2

GA operators: Selection

- Main idea: better individuals get higher chance
  - Chances proportional to fitness
  - Implementation: roulette wheel technique
    - » Assign to each individual a part of the roulette wheel
    - » Spin the wheel n times to select n individuals



←

fitness(A) = 3

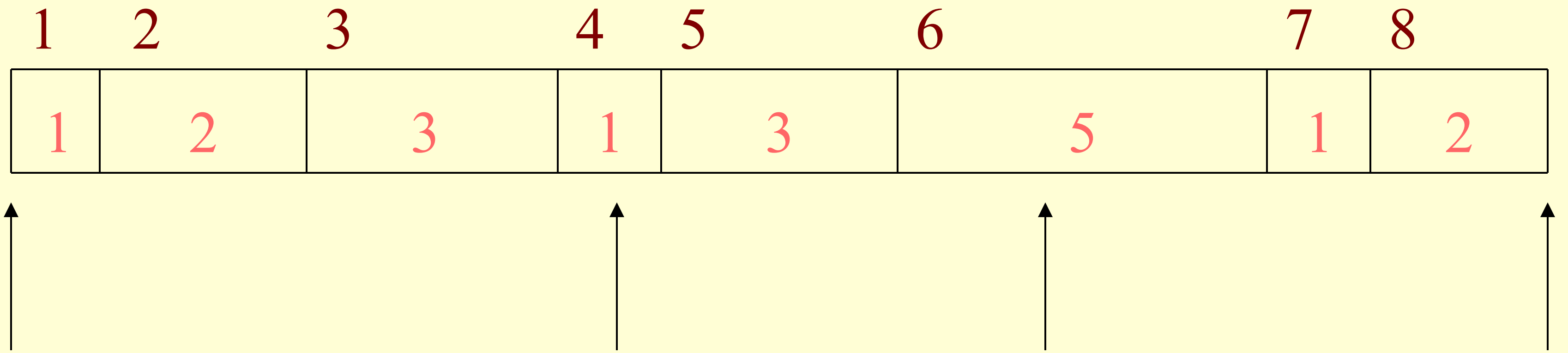
fitness(B) = 1

fitness(C) = 2

GA operator: Selection

- – Add up the fitness's of all chromosomes
- – Generate a random number R in that range
- – Select the first chromosome in the population that -when all previous fitness’s are added including the current one- gives you at least the value R

Roulette Wheel Selection



1	2	3	4	5	6	7	8
1	2	3	1	3	5	1	2

0

Rnd[0..18] = 7

Rnd[0..18] = 12

18

Chromosome4

Chromosome6

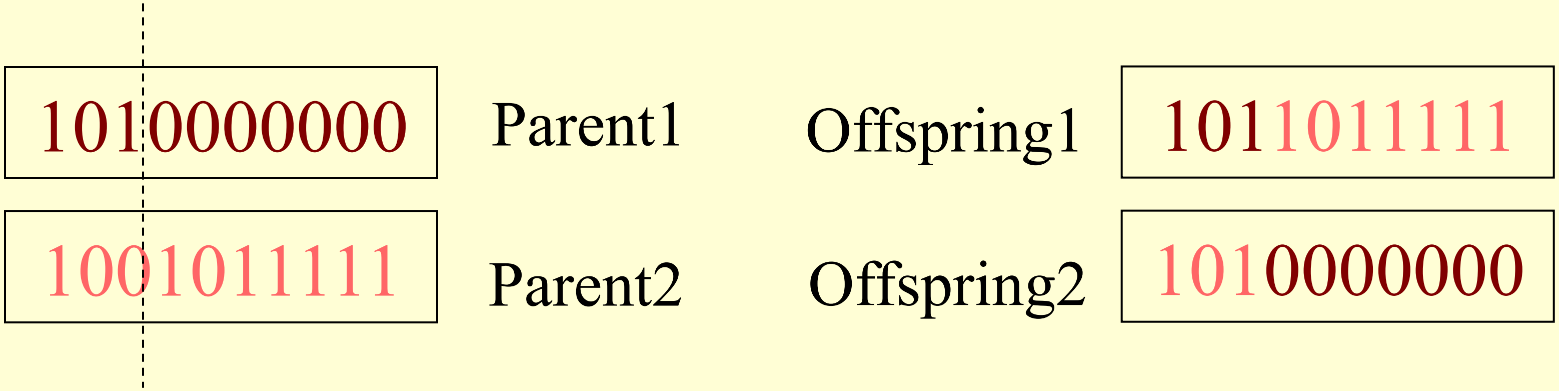
Parent1

Parent2

GA operator: Crossover

- Choose a random point on the two parents
- Split parents at this crossover point
- With some high probability (*crossover rate*) apply crossover to the parents
  - P_c typically in range (0.6, 0.9)
- Create children by exchanging tails

Crossover - Recombination



1010000000 Parent1

1001011111 Parent2

Offspring1 1011011111

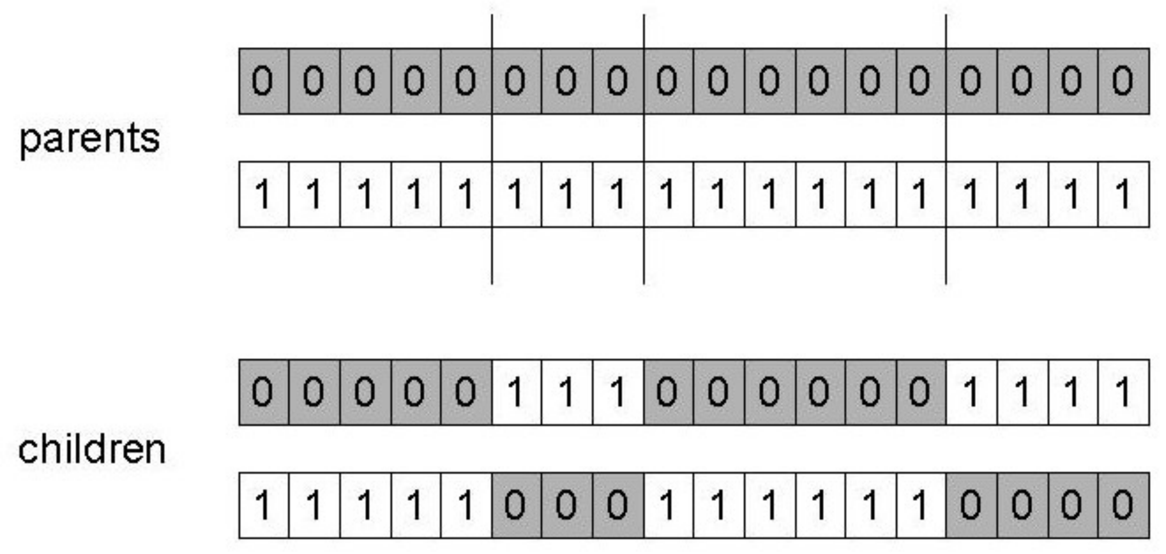
Offspring2 1010000000

Crossover single point - random

Single Point Crossover

n-point crossover

- Choose n random crossover points
- Split along those points
- Glue parts, alternating between parents
- Generalisation of 1 point (still some positional bias)



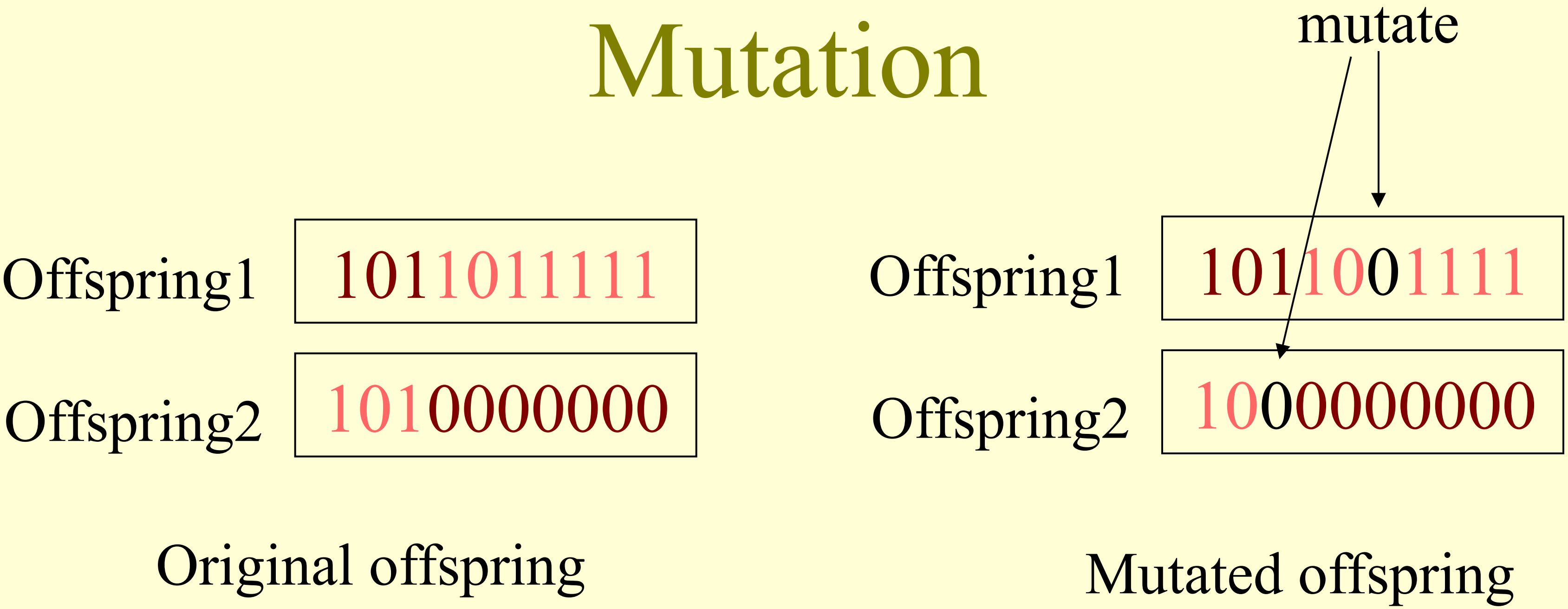
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 1 1 1

1 1 1 1 1 0 0 0 1 1 1 1 1 1 0 0 0 0

Mutation



mutate

Offspring1 1011011111

Offspring2 1010000000

Original offspring

Offspring1 1011001111

Offspring2 1000000000

Mutated offspring

With some small probability (the *mutation rate*) flip each bit in the offspring (*typical values between 0.1 and 0.001*)

Ensemble: Single Objective Optimization

Weighted Vote based Classifier Ensemble

- Motivation
  - All classifiers are not equally good to identify all classes

- Weighted voting: weights of voting vary among the classes for each classifier
  - *High*: Classes for which the classifier perform good
  - *Low*: Classes for which it's output is not very reliable
- *Crucial issue*: Selection of appropriate weights of votes per classifier

Problem Formulation

Let *no. of classifiers*=N, and *no. of classes*=M

Find the weights of votes V per classifier optimizing a function F(V)

-V: an real array of size N × M

-V(i , j) : weight of vote of the *i*th classifier for the *j*th class

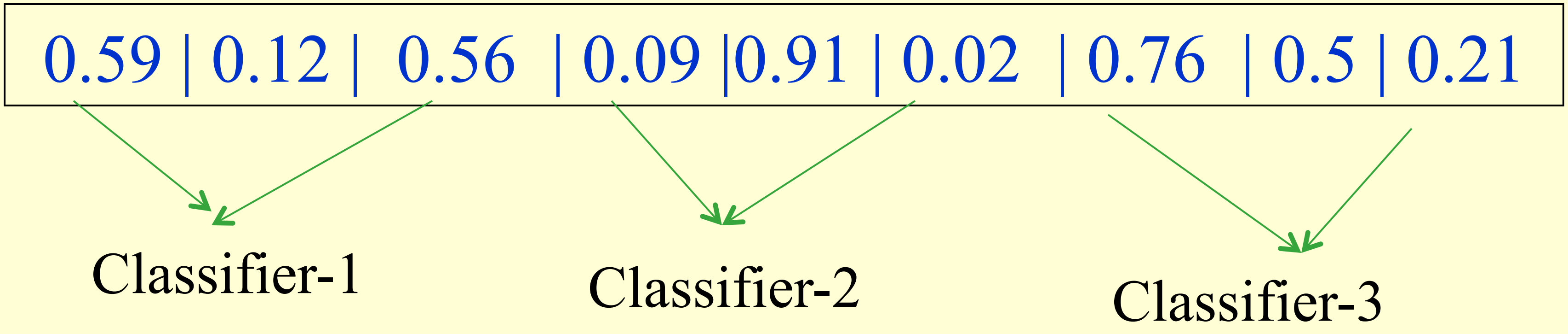
-V(i , j) ε [0, 1] denotes the degree of confidence of the *i*th classifier for the *j*th class

maximize F(B) ;

F ε *{recall, precision, F-measure}* and B is a subset of A

Here, *F1= F-measure*

Chromosome representation



0.59 | 0.12 | 0.56 | 0.09 |0.91 | 0.02 | 0.76 | 0.5 | 0.21

Classifier-1

Classifier-2

Classifier-3

- Real encoding used
- Entries of chromosome randomly initialized to a real (r) between 0 and 1: r = rand () / RAND_MAX+1
- If the population size P then all the P number of chromosomes of this population are initialized in the above way

Fitness Computation

Step-1: For M classifiers, F_i *i*= 1 to M be the F-measure values

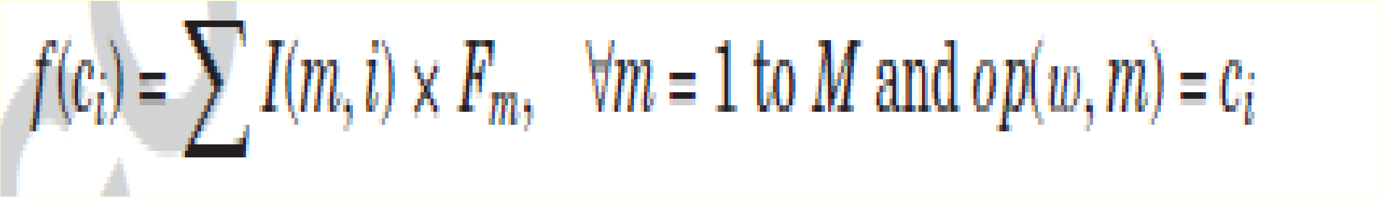
Step-2: Train each classifier with 2/3 training data and test with the remaining 1/3 part

Step-3: For ensemble output of the 1/3 test data, apply weighted voting on the outputs of M classifiers

(a). Weight of the output label provided by the *ith* classifier = I (m, i)

Here, *I(m, i) is the entry of the chromosome corresponding to mth* classifier and *ith class*

(b). Combined score of a class for a word *w*



$$f(c_i) = \sum I(m, i) \times F_m, \quad \forall m = 1 \text{ to } M \text{ and } op(w, m) = c_i$$

Fitness Computation

Op(w, m): output class produced by the *mth* classifier for word *w*

Class receiving the maximum score selected as joint decision

Step-4: Compute overall F-measure value for 1/3 data

Step-5: Steps 3 and 4 repeated to perform 3-fold cross validation

Step-6: Objective function or fitness function = F-measure$_{avg}$

Objective: Maximize the objective function using search capability of GA

Other Parameters

- Selection
  - Roulette wheel selection (*Holland, 1975; Goldberg, 1989*)
- Crossover
  - Normal Single-point crossover (Holland, 1975)
- Mutation
  - Probability selected adaptively (*Srinivas and Patnaik, 1994*)
  - Helps GA to come out from local optimum

Mutation

- Each position in a chromosome mutated with probability μ_m in the following way
  - Value replaced with a random variable drawn from a *Laplacian distribution*

$$p(\epsilon) \propto e^{-\frac{|\epsilon - \mu|}{\delta}}$$

δ : scaling factor; sets the magnitude of perturbation

μ : value at the position to be perturbed

Scaling factor $\delta = 0.1$

Termination Condition

- Execute the processes of *fitness computation*, *selection*, *crossover*, and *mutation* for a maximum number of generations
- *Best solution*-Best string seen up to the last generation

- Best solution indicates
  - Optimal voting weights for all classes in each classifier

- Elitism implemented at each generation
  - Preserve the best string seen up to that generation in a location outside the population
  - Contains the most suitable classifier ensemble

NE Extraction in Biomedicine

- Objective-identify biomedical entities and classify them into some predefined categories
  - *E.g. Protein*, *DNA*, *RNA*, *Cell_Line*, *Cell_Type*
- *Major Challenges*
  - building a complete dictionary for all types of biomedical NEs is infeasible due to the generative nature of NEs
  - NEs are made of very long compounded words (i.e., contain nested entities) or abbreviations and hence difficult to classify them properly
  - names do not follow any nomenclature

Challenges (Contd..)

- NEs include different symbols, common words and punctuation symbols, conjunctions, prepositions etc.
  - NE boundary identification is more difficult and challenging

- Same word or phrase can refer to different NEs based on their contexts

Features

- Context Word: Preceding and succeeding words
- Word Suffix and Prefix
    - Fixed length character strings stripped from the ending or beginning of word
- Class label: Class label(s) of the previous word (s)
- Length (binary valued): Check whether the length of the current word less than three or not (shorter words rarely NEs)
- Infrequent (binary valued): Infrequent words in the training corpus most probably NEs

Features

- Part of Speech (PoS) information- PoS of the current and/or surrounding token(s)
    - GENIA tagger V2.0.2 (http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/tagger)
- Chunk information-Chunk of the current and/or surrounding token(s)
    - GENIA tagger V2.0.2
- Unknown token feature-checks whether current token appears in training

Features

- Word normalization
  - feature attempts to reduce a word to its stem or root form (from GENIA tagger O/P)

- Head nouns
  - major noun or noun phrase of a NE that describes its function or the property
  - E.g. *factor* is the head noun for the NE *NF-kappa B transcription factor*

Features

- Verb trigger-special type of verb (e.g., *binds*, *participates* etc.) that occur preceding to NEs and provide useful information about the NE class

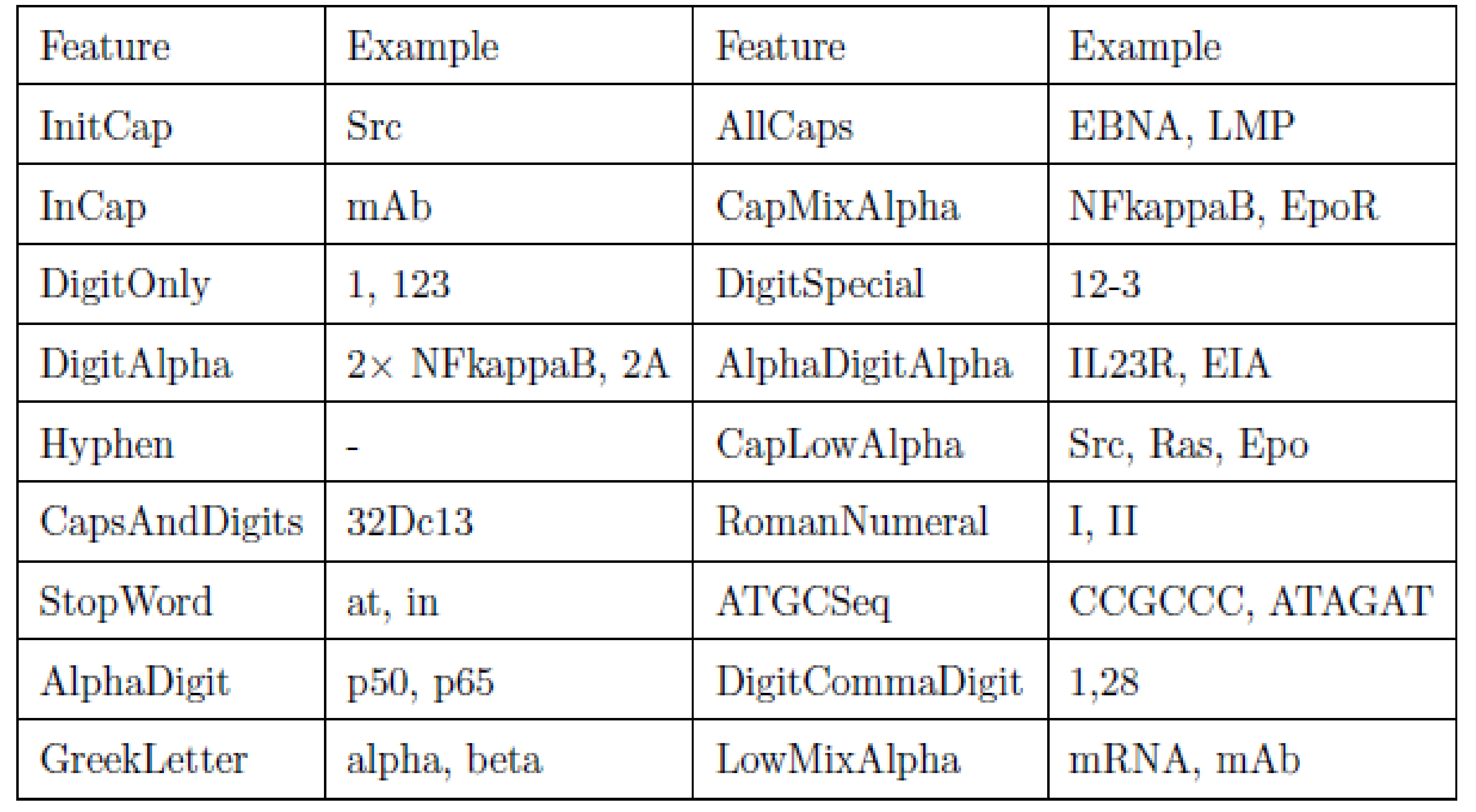
- Word class feature-Certain kinds of NEs, which belong to the same class, are similar to each other
  - capital letters→ A, small letters→a, number→O and non-English characters→-
  - consecutive same characters are squeezed into one character
  - groups similar names into the same NE class

Features

- Informative words
  - NEs are two long, complex and contain many common words that are actually not NEs
  - Function words- *of*, *and* etc.; nominals such as *active*, *normal* etc. appear in the training data often more frequently but these don't help to recognize NEs
  - Feature extracts informative words from training data statistically
- Content words in surrounding contexts-*Exploits global context information*

Features

- *Orthographic Features*-number of orthographic features depending upon the contents of the wordforms



Feature	Example	Feature	Example
InitCap	Src	AllCaps	EBNA, LMP
InCap	mAb	CapMixAlpha	NFkappaB, EpoR
DigitOnly	1, 123	DigitSpecial	12-3
DigitAlpha	2× NFkappaB, 2A	AlphaDigitAlpha	IL23R, EIA
Hyphen	-	CapLowAlpha	Src, Ras, Epo
CapsAndDigits	32Dc13	RomanNumeral	I, II
StopWord	at, in	ATGCSeq	CCGCCC, ATAGAT
AlphaDigit	p50, p65	DigitCommaDigit	1,28
GreekLetter	alpha, beta	LowMixAlpha	mRNA, mAb

Experiments

- Datasets-JNLPBA 2004 shared task datasets
  - Training: 2000 MEDLINE abstracts with 500K wordforms
  - Test: 404 abstracts with 200K wordforms
- Tagset: 5 classes
  - Protein, DNA, RNA, Cell_line, Cell_type
- Classifiers
  - CRF and SVM

- Evaluation scheme: JNLPBA 2004 shared task script (http://www-tsujii.is.s.u-tokyo.ac.jp/GENIA/ERtask/report.html)
  - Recall, precision and F-measure according to *exact boundary match*, *right* and *left* boundary matching

Experiments

Model	Recall	Precision	F-measure
Best individual classifier	73.10	76.76	74.76
Baseline-1	71.03	75.76	73.32
Baseline-II	71.42	75.90	73.59
Baseline-III	71.72	76.25	73.92
SOO based ensemble	74.17	77.87	75.97

- Baseline-I: Simple majority voting of the classifiers
- Baseline-II: Weighted voting where weights are based on the overall F-measure value
- Baseline-III: Weighted voting where weights are the F-measure of the individual classes

Issues of Cross-corpus Compatibilities

- No unified annotation scheme exists for the biomedical entity annotation

- Building a system that performs reasonably well for almost all the domain is important!

- Datasets used in the experiments
  - JNLPBA shared task datasets
  - GENETAG datasets
  - AIMed datasets
- Differ in text selection as well as annotation

GENIA: Properties

- GENIA Version 3.02 corpus of the GENIA project
  - http://research.nii.ac.jp/collier/workshops/JNLPBA04st.htm
- Constructed by a controlled search on Medline using MeSH terms such as *human, blood cells and transcription factors*
- Originally annotated with a taxonomy consisting of 48 classes
  - Converted to 5 classes for the shared task data
  - Protein, DNA, RNA, Cell_line, Cell_type
- No embedded structures

GENIA: Properties

- *protein* annotation was applied only to proteins, while genes were annotated in the scope of *DNA* annotations

- Word "*protein*" included as part of protein name almost always

- Test data- Super domain of blood cells and transcription factors

AIMed: Properties

- Focuses on the human domain, and exhaustively collect sentences from the abstracts of PubMed

- Word "protein" is not always included as part of protein name
  - Boundary ambiguities thus degrade system performance
- Unlike GENIA protein families are not annotated

- Only specific names that could ultimately be traced back to specific genes in the human genome are tagged
  - For example,"*tumor necrosis factor*" was not annotated while "*tumor necrosis factor alpha*" was tagged

- Annotations in AIMed include some gene names without differentiating them from proteins

GENETAG: Properties

- Unlike GENIA and AIMed , GENETAG covers a more general domain of PubMed

- Contains both true and false gene or protein names in a variety of contexts

- Not all the sentences of abstracts were included, rather more NE informative sentences were considered

- In terms of text selection, GENIA and GENETAG are closer to each other, compared to AIMed

- Like GENIA, GENETAG also includes the semantic category word ‘protein’ for protein annotation

Experimental Setups

- Experimental Setup-I:
  - GENIA corpus by replacing all tags except ‘Protein’ by ‘O’ (other-than-NE) + AIMed corpus
  - Cross-validation
- Experimental Setup-II:
  - ‘Protein’ and ’DNA’ annotations of GENIA+ Replace all other annotations by ‘O’+ AIMed corpus
  - Cross-validation

Experiments

- Experimental Setup-III:
  - GENIA corpus by replacing all tags except ‘Protein’ by ‘O’ (other-than-NE) + GENETAG corpus
  - Test on GENETAG

- Experimental Setup-IV:
  - GENIA with only ‘Protein’, ‘DNA’ and ‘RNA’ annotations + GENETAG corpus
  - Test on GENETAG corpus

Results: Cross Corpus

Approach	Training set	Test set	Recall	Precision	F-measure
Best Ind. Classifier	JNLPBA (protein only) +AIMed	AIMed	83.14	83.19	83.17
SOO	JNLPBA (protein only) +AIMed	AIMed	85.10	85.01	85.05
Best Ind. Classifier	JNLPBA (protein + DNA) +AIMed	AIMed	82.17	84.15	83.15
SOO	JNLPBA (protein + DNA) +AIMed	Cross validation	84.07	86.01	85.03
Best Ind. Classifier	JNLPBA (protein only) +GENETAG	GENETAG	89.44	93.07	91.22
SOO	JNLPBA (protein only) +GENETAG	GENETAG	91.19	94.98	93.05
Best Ind. Classifier	JNLPBA (protein+DNA +RNA)+GENTAG	GENETAG	88.70	93.55	91.06
SOO	JNLPBA (protein+DNA +RNA)+GENTAG	GENETAG	90.09	95.16	92.56

Results: Original Datasets

Dataset	Model	Recall	Precision	F-measure
GENIA	Best individual classifier	73.10	76.78	74.90
	SOO	74.17	77.87	75.97
AIMed	Best individual classifier	94.56	92.66	93.60
	SOO	95.65	94.23	94.93
GENETAG	Best individual classifier	95.35	95.31	95.33
	SOO	95.99	95.81	95.90

Drop in performance by around 10% for AIMed
and around 3% for GENETAG

Why MOO in Classifier Ensemble?

- Single objective optimization technique : optimizes a single quality measure
  - recall, precision or F-measure at a time
- A single measure cannot capture the quality of a good ensemble reliably
- A good classifier ensemble should have it's all the parameters optimized simultaneously

- Advantages of MOO
  - MOO to simultaneously optimize more than one classification quality measures
  - Provides user a set of alternative solutions

Formulation of Classifier Ensemble Selection Problem

Classifier ensemble selection problem:

A: *Set of N classifiers*

Find a set of classifiers B that maximizes

[$F1(B)$, $F2(B)$]

where

$F1, F2 \in \{recall, precision, F\text{-}measure\}$ *and*

$F1 \neq F2$

Here, $B \subseteq A$

$F1 = recall$ and $F2 = precision$

Classifier Ensemble Selection: Proposed Approach

Chromosome representation

010110111110011111

Total number of available classifiers: M

0 at position *i*- *i*th classifier does not participate in ensemble

1 at position *i* – *i*th classifier participates in ensemble

Fitness Computation

Step-1: For M classifiers, F_i i= 1 to M be the F-measure values

Step-2: Train each classifier with 2/3 training data and test with the remaining 1/3 part.

Step-3: For ensemble output of the 1/3 test data

a. Appropriate class is determined from the weighted voting

b. weight = F-measure value of the respective classifier

Step-4: Calculate the overall *recall*, *precision* and *F-measure* values for 1/3 data

Steps 2 -4 are repeated 3 times to perform 3-fold cross validation.

Step-5: Average *recall* and *precision* values are considered as two objective functions

Other Operators

- Steps of non-dominated sorting genetic algorithm (NSGA-II) are executed (Deb K et al., 2002)
- Crowded binary tournament selection
- Conventional *crossover* and *mutation*

- *Elitism*-non-dominated solutions among the parent and child populations are propagated to the next generation (Deb K, 2001)

- *Near-Pareto-optimal* strings of the last generation provide the different solutions to the ensemble problem

Selecting Solution from Pareto Optimal Front

- In MOO, the algorithms produce a *large number of non-dominated solutions* on the final *Pareto optimal front*
- Each of these solutions provides a classifier ensemble
- All the *solutions are equally important* from the algorithmic point of view
- User may want only a single solution

Selecting Solution from Pareto Optimal Front

- For every solution on the final Pareto optimal front
  - calculate the overall average *F-measure value of the classifier ensemble* for the three-fold cross-validation

- Select the solution with the maximum *F-measure value* as the best solution

- Evaluate the classifier ensemble corresponding to the best solution on the test data

Experiments

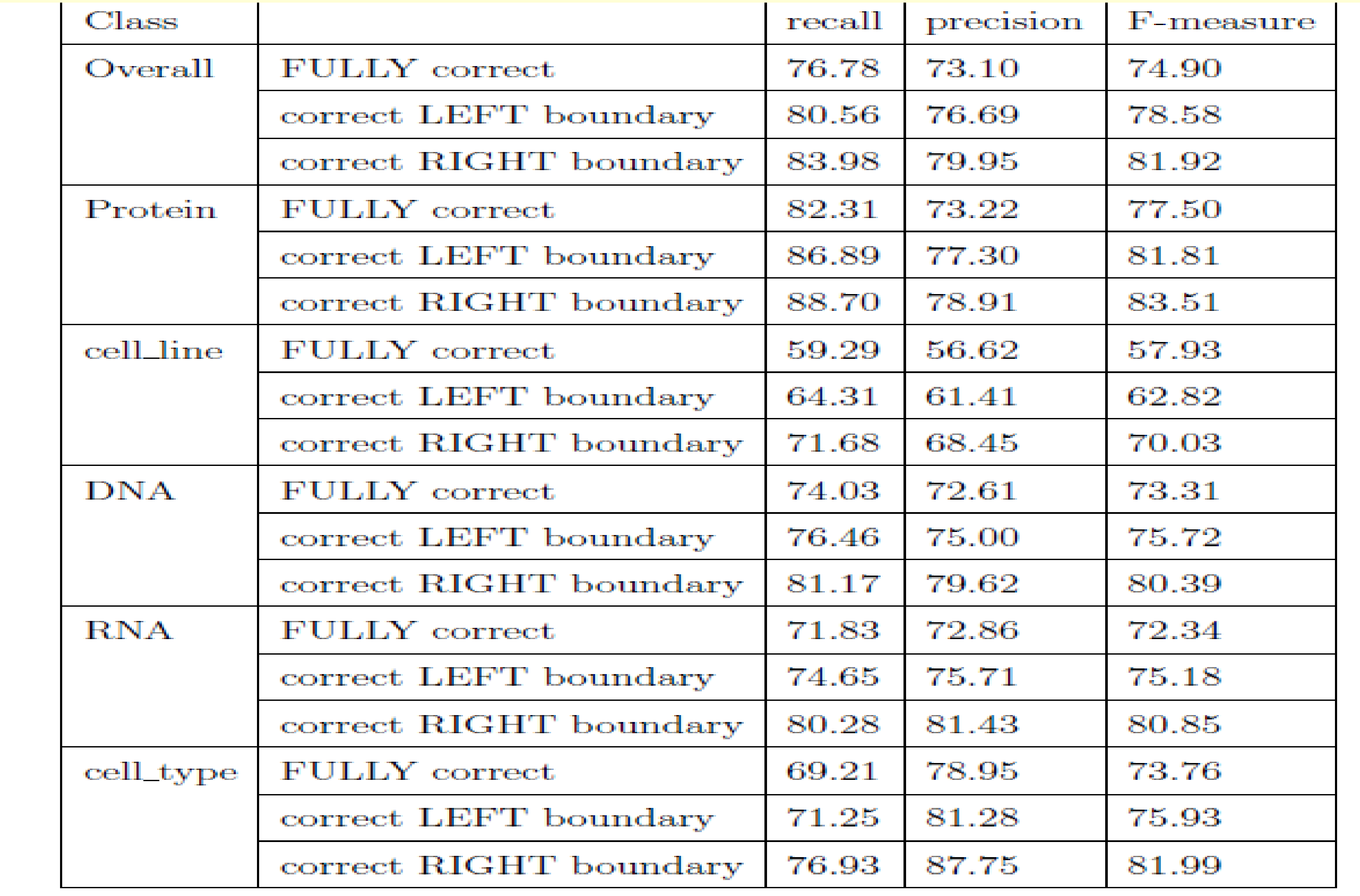
- Base classifiers
  - Based on different feature representations, several CRF and SVM classifiers built
- Objective functions (*in this work*)
  1. MOO1: overall average *recall* and *precision*
  2. MOO2: average F-measure value of five classes
  3. MOO3: average recall and precision values of five classes
  4. MOO4: average F-measure values of individual NE boundaries

Experiments (Results)

Model	Recall	Precision	F-measure
Best individual classifier	73.10	76.78	74.90
MOO1	75.52	78.03	76.75
MOO2	75.78	78.45	77.09
MOO3	75.91	78.98	77.41
MOO4	76.15	79.09	77.59

Around 2% improvement over the present state-of-the-art

Experiments



Class		recall	precision	F-measure
Overall	FULLY correct	76.78	73.10	74.90
	correct LEFT boundary	80.56	76.69	78.58
	correct RIGHT boundary	83.98	79.95	81.92
Protein	FULLY correct	82.31	73.22	77.50
	correct LEFT boundary	86.89	77.30	81.81
	correct RIGHT boundary	88.70	78.91	83.51
cell_line	FULLY correct	59.29	56.62	57.93
	correct LEFT boundary	64.31	61.41	62.82
	correct RIGHT boundary	71.68	68.45	70.03
DNA	FULLY correct	74.03	72.61	73.31
	correct LEFT boundary	76.46	75.00	75.72
	correct RIGHT boundary	81.17	79.62	80.39
RNA	FULLY correct	71.83	72.86	72.34
	correct LEFT boundary	74.65	75.71	75.18
	correct RIGHT boundary	80.28	81.43	80.85
cell_type	FULLY correct	69.21	78.95	73.76
	correct LEFT boundary	71.25	81.28	75.93
	correct RIGHT boundary	76.93	87.75	81.99

OPTIMIZING COREFERENCE

Joint work with

SriparnaSaha (IIT Patna)
Olga Uryupina (Uni Trento)
Massimo Poesio (UniEssex / Uni Trento)

Coreference Metrics

- MUC SCORE (Vilain et al, 1995)
  - Measures precision / recall over LINKS
- B3 (Bagga& Baldwin, 1998)
  - Measures precision / recall over MENTIONS
- CEAF (Luo, 2005)
  - Measures precision / recall over ENTITIES
- MELA (Pradhan et al 2011)
  - Average of other metrics

THE PROBLEM WITH COREFERENCE METRICS

SEMEVAL 2010: EN, SYSTEM MARKABLES

SYSTEM	MUC R	MUC P	MUC F	CEAF R	CEAF P	CEAF F
BART	**62.8**	**52.4**	**57.1**	**70.1**	**64.3**	**67.1**
CORRY-B	54.7	55.5	55.1	70.9	67.9	69.4
CORRY-M	61.5	53.4	57.2	66.3	64.5	64.8

Proposals/Considerations

- Multiple evaluation metrics exist
  - Not only for coreference

- Could take advantage of them by optimizing using a Multiobjective Function

Aspects that we Optimized

- ARCHITECTURE
  - NO-SPLIT / PRO-NONPRO / PRO-DET-NAM / etc
- Type of CLASSIFIER
  - MAXENT / SVM / DECISION TREE
- FEATURES
- PARAMETERS for the classifiers

Features (~ 50 IN TOTAL)

- MENTION TYPE/SUBYTPE
  - 7 in total, fine/coarse classification of Mi and Mj
- AGREEMENT
  - GENDER, NUMBER, ANIMACY, SEMCLASS
- ALIASING
- SYNTAX (FROM PARSE TREES)
  - APPOSITIVE, COORDINATION, COPULA, SYNDEPTH, ….
- MATCHING (STRING MATCH, HEAD MATCH …)
- SALIENCE (FIRST MENTION, …)
- WEB (SAME WIKIPEDIA / YAGO ENTRY, PATTERNS)
- PROXIMITY (DISTANCE IN MARKABLES, SENTENCES, ..)
- MISC (SPEAKER ALIAS, …)

Parameters

- Decision Trees:
  - Confidence value (75%, 50%, 25%, … 1%)
  - Minimum number of splitoff (2 to 5)
- SVM:
  - Kernel function: linear, polynomial, radial basis, sigmoid
- MaxEnt:
  - Regularization
  - Composite features

Example of Chromosome

pro-nonpro
Decision-tree 0.25 3
01110101010101000011010111000010101111001111
SVM 2.00
11010000111111101110100010101001010010001111

Operators

- 4 Mutation Operations:
  - Change of split, Classifier, Features, Parameters
- Crossover
- Selection:
  - SOO: Random
  - MOO: Crowded binary tournament (Deb et al 2002)

The Metrics

- F_{MUC} : MUC SCORE (VILAIN ET AL 1995)
- F_{CEAFE}: CEAF, ENTITY ALIGNMENT (LUO 2005)
- F_{CEAFM}: CEAF, MENTION ALIGNMENT (LUO)
- F_{B3}: B3 (BAGGA&BALDWIN 1999)

- F_{MELA}: CONL-11 Composite measure (Pradhan et al., 2011): used to choose between elements of the pareto front

Experimental Setup

- Anaphora resolution platform: BART (Versley et al, 2008)
  - Open-source, modular toolkit supporting a number of architectures / classifiers
- Anaphora resolution model: MENTION-PAIR
- Datasets:
  - ACE02
  - OntoNotes 3.0

RESULTS: BASELINES

		F_MUC	F_B3	F_CEAFm	F_CEAFE
gbnews	Soon et al 2001	71.43	74.29	68.15	71.42
	All features	73.70	73.16	68.29	72.49
	State-of-the-art	65-69			
gnpaper	Soon et al 2001	71.05	71.43	65.45	68.58
	All features	71.65	69.15	63.62	65.46
	State-of-the-art	70-72			
gnwire	Soon et al 2001	69.40	75.39	69.12	71.35
	All features	72.44	75.96	71.26	71.82
	State-of-the-art	54-67			
cbnews	Soon et al 2001	60.63	71.09	60.41	61.23
	All features	61.73	69.88	59.79	59.92
cnpaper	Soon et al 2001	65.56	70.18	61.68	61.29
	All features	64.62	66.45	55.97	56.38
cnwire	Soon et al 2001	61.33	72.19	62.74	64.29
	All features	63.41	70.62	61.18	62.17

RESULTS: MOO(FS) VS MOO(FS,PS)

	F_MELA	F_MUC	F_B3	F_CEAFm	F_CEAFE
MOO(FS)					
gnpaper	69.65	71.31	70.31	64.85	67.33*
cnpaper	64.74*	65	69.42*	61.06*	59.82*
gnwire	73.04	71.62	75.53	69.77	71.99
cnwire	66	62.39	72.38	62.73	63.25
cbnews	65.17	62.21	72.05*	61.20*	61.27
gbnews					
MOO(FS,PS)					
gnpaper	71.55*+	72.24	72.17*+	66.94*+	70.26*+
cnpaper	67.20*+	67.49*+	71.93*+	64.50*+	62.18*+
gnwire	74.86+	72.95	77.01+	72.45+	74.62+
cnwire	67.25	63.62	73.65*+	64.58*	64.5*+
cbnews	66.40*+	64.16*+	73.14*+	62.74*+	61.92*
gbnews					

Conclusions

- Evolutionary optimization can be effectively used for different NLP problems

- MOO performs often superior compared to SOO

- Proposed method is general enough to be applicable for other domains

References

Asif Ekbal and Sriparna Saha (2013). Stacked ensemble coupled with feature selection for biomedical entity extraction, ***Knowledge Based Systems***, volume (46), PP. 22–32, Elsevier.

S.Saha, A. Ekbal and U. Sikdar (2013). Named Entity Recognition and Classification in Biomedical Text Using Classifier Ensemble. International Journal on Data Mining and Bioinformatics (in press).

A. Ekbal and S. Saha (2010). Classifier Ensemble Selection Using Genetic Algorithm for Named Entity Recognition. Research on Language and Computation (RLC), Vol. (8), PP. 73-99, Springer

A. Ekbal and S. Saha (2012). Multiobjective Optimization for Classifier Ensemble and Feature Selection: An Application to Named Entity Recognition. International Journal on Document Analysis and Recognition (IJDAR), Vol. 15(2), 143-166, Springer

A.Ekbal and S. Saha (2011). Weighted Vote-Based Classifier Ensemble for Named Entity Recognition: A Genetic Algorithm-Based Approach. ***ACM Transactions on Asian Language Information Processing (ACM TALIP***), Vol. 2(9), ACM, DOI = 10.1145/1967293.1967296 http://doi.acm.org/10.1145/1967293.1967296.

References

- S. Saha, **A. Ekbal**, O. Uryupina, M. Poesio (2011). Single and multi-objective optimization for feature selection in anaphora resolution. *5th International Joint Conference on Natural Language Processing **(IJCNLP 2011)***, Thailand, November 8-13, PP. 93-101.

- **A. Ekbal**, S. Saha, O. Uryupina, M. Poesio (2011). Multiobjective Simulated Annealing Based Approach for Feature Selection in Anaphora Resolution, ***Lecture Notes in Computer Science***, Volume 7099, 2011, pp: 47-58.

Thank you for your attention!