Evolutionary Learning of Syntax Patterns for Genic Interaction Extraction

Alberto Bartoli, Andrea De Lorenzo, Eric Medvet,
Fabiano Tarlao, Marco Virgolin

MACHINE LEARNING LAB

UNIVERSITÀ DEGLI STUDI DI TRIESTE
DIPARTIMENTO DI INGEGNERIA E ARCHITETTURA

Problem

➔ Identifying sentences that contain **interactions** between **genes** and **proteins**
- ◆ from biomedical literature

➔ Available data:
- ◆ dictionary of genes, proteins and interactors
- ◆ example sentences

Why?

- ➔ Biomedical literature is:
  - vast
  - rapidly growing
- ➔ Challenging problem: automatic extraction of knowledge from a text in natural language
  - informations are “diluted” in the text
  - very challenging problem: **discover relations** between entities

MACHINE LEARNING LAB

Goal

➔ Generation of a classifier C in order to identify sentences containing interactions between genes and proteins

- **automatically**
- based on recurring **syntactic patterns**

Our approach

- Classifier C is a set of **regular expressions** (*regex*)

$$C=\{r_1,r_2,\dots\}$$

- Each regex is a sentence classifier ("accepts" or "does not accept")
  - C accepts sentences accepted by **at least one** regex

- Regex applied on a **semantical representation** of the text

Our approach (II)

➔ Regex generated **automatically**

- by means of Genetic Programming (GP)
- starting from **examples**
  - strings which must be accepted
  - strings which must **not** be accepted

MACHINE LEARNING LAB

Sentences preprocessing

Mapping of a sentence s in a ϕ-string x

a. substitution of words in s with "**annotations**"
   i. gene, protein, interactor

   or

   ii. Part-Of-Speech

b. **mapping** of annotations in Unicode characters

c. concatenation

MACHINE LEARNING LAB

Sentences preprocessing (II)

Example:

S = **YfhP** may act as a negative **regulator** for the **transcription** of **yfhQ**

↓

[**YfhP**] [may] [act] [as] [a] [negative] [**regulator**] [for] [the] [**transcription**] [of] [**yfhQ**]

↓

[**GENEPTN**] [MD] [VB] [IN] [DT] [JJ] [**INOUN**] [IN] [DT] [**INOUN**] [IN] [**GENEPTN**]

↓

x = GB0if6JifJiG

Generation of C: GP

➔ We used a Tree-based GP

➔ In this work
candidate solution = regex

Key aspects

- ➔ Multi-objective fitness:
  - ◆ $f=(Accuracy, FPR, Regex\ length)$
  - ◆ we purposefully avoided to include any problem-specific knowledge (gene/protein/…)
- ➔ Problem handled by mean of *separate-and-conquer*
- ➔ Final output: set of regular expressions $C=\{r_1, r_2,\ldots\}$

Separate-and-conquer

- ➔ Each regex $r_i \in C$ makes an **independent** and **parallel** classification
- ➔ Each regex is tailored for a sub-problem
  - ◆ the problem is solved “step-by-step”
- ➔ Final output = **logic OR** of classifications

Separate-and-conquer

- $C = \varnothing$
- we execute a GP search over the examples obtaining r^*
- if $FPR < threshold$
  - $C = C \cup \{r^*\}$
- else
  - terminate
- remove from the positive examples those which were classified correctly by r^*

MACHINE LEARNING LAB

Classifier example

$$C = \{r_1, r_2\}$$

r_1 = `GENEPTN[^RB][^NNS VBN GENEPTN]++`

r_2 = `. INOUN IN GENEPTN . [^DT NN]`

Experimental evaluation: the data

- Dataset: 456 sentences from biomedical papers
  - ½ with interactions e ½ without
  - manually labelled by experts
- Dataset splitted in *Learning* e *Testing*
  - ≈80% examples in Learning
  - ≈20% examples in Testing
- 5 fold randomly generated
  - with $Testing_i \neq Testing_j$

Baseline 1, 2: *problem specific knowledge*

- ➔ *Annotations-Co-Occurrence*
  - it is tightly tailored to this specific problem
  - sentence is positive if contains
    - at least 2 genes/proteins
    - at least 1 interactor
- ➔ *Annotations-LLL05-Patterns*
  - 10 pattern generated in *"LLL'05 Challenge: Genic Interaction Extraction with Alignments and Finite State Automata"*
    - J. Hakenberg et alia
  - built over ***>90%* of the dataset** (also testing!)

MACHINE LEARNING LAB

Baseline 3: ϕ-SSLEA

- Based on *Smart State Labeling Algorithm*
  - algorithm for *DFA learning*
  - works well in presence of noise
- *Hill-Climbing*
- Generates **DFA** which accepts or refuse a ϕ-string x
  - if x accepted $\Rightarrow$ x contains an interaction between gene/protein
  - otherwise, no

Baseline 4, 5: *Words-NaiveBayes* e *Words-SVM*

- Standard for text classification
  - Supervised Machine Learning methods
- Feature based on **word occurrences**
- Preprocessing
  - stemming
  - features selection

Results

Averaged over the 5 folds

Classifier	Accuracy	FPR	FNR
Annotations-Co-Occurrence	77.8	40.0	4.5
Annotations-LLL05-Patterns	82.3	25.0	10.5
Words-NaiveBayes	51.3	25.0	95.0
Words-SVM	73.8	29.0	23.5
ϕ-SSLEA	59.8	44.0	33.5
C	**73.7**	**23.5**	**22.5**

Results (II)

- ➔ C performs as well as Word-SVM and better than other learning approaches
- ➔ accuracies of C and *Annotations-Co-Occurrence* (which exploits domain knowledge of an expert) are **very close**
  - Pro: C is composed by patterns (regex) readable
  - Con: time to generate C (hours) $\gg$ time to generate other methods (minutes)
    - but ≈ time taken for classifying (seconds)

Conclusions

We proposed:

➔ a method for the **automatic synthesis** of a classifier for **natural language** sentences
- based on **syntactic pattern**
- by mean of **GP**
- **separate-and-conquer**

➔ results are highly promising