Collocation Extraction using Genetic Programming

Bojana Dalbelo Bašić

University of Zagreb
Faculty of Electrical Engineering and Computing
bojana.dalbelo@fer.hr,

Worksp KDSA 2009
Zagreb, IRB
2009-11-12

- Jan Šnajder, Bojana Dalbelo Bašić, Saša Petrović, Ivan Sikirić, *Evolving new lexical association measures using genetic programming,* The 46th Annual Meeting of the **A**ssociation of **C**omputational **L**inguistic: Human Language Technologies, Columbus, Ohio, June 15-20, 2008.

Outine

- Collocations
- Genetic programming
- Results
- Conclusion

Collocation

- (Manning and Schütze 1999)

  "...an expression consisting of two or more words that correspond to some conventional way of saying things."

- Many different deffinitions ...

- An uninterrupted sequence of words that generally functions as a single constituent in a sentence (e.g., *stock market*, *Republic of Croatia*).

Collocation

Applications:

- improving indexing in information retrieval (Vechtomova, Robertson, and Jones 2003)
- automatic language generation (Smadja and McKeown 1990)
- word sense disambiguation (Wu and Chang 2004),
- terminology extraction (Goldman and Wehrli 2001)
- improving text categorization systems (Scott and Matwin 1999)

Collocation

More general term - *n*-gram of words – any sequence of *n* words (digram, trigram, tetragram)

Collocation extraction is usually done by assigning each candidate n-gram a value indicating how strongly the words within the n-gram are associated with each other.

Collocation extraction

More general term - *n*-gram of words – any sequence of *n* words (digram, trigram, tetragram)

Collocation extraction is usually done by assigning each candidate n-gram a value indicating how strongly the words within the n-gram are associated with each other.

Association measures

Association measures

Examples:

- MI (*Mutual Information*):

$$I(a,b) = \log_2 \frac{P(ab)}{P(a)P(b)}$$

- DICE coefficient:

$$DICE(a,b) = \frac{2f(ab)}{f(a) + f(b)}$$

Association measures

Based on hypothesis testing:

- **χ^2:**

$$\chi^2 = \sum_{i,j} \frac{(O_{ij} - E_{ij})^2}{E_{ij}}$$

- **log-likelihood:**

$$G^2 = \sum_{i,j} O_{ij} \log \frac{O_{ij}}{E_{ij}}$$

Collocation extraction

Example:

digram	Assoc.measure value
stock market	20.1
machine learning	30.7
town Slavonski	10.0
New York	25.2
big dog	7.2
new house	7.4
White house	16.2

Collocation extraction

Example:

digram	Assoc.measure value
machine learning	30.7
New York	25.2
stock market	20.1
White house	16.2
town Slavonski	10.0
new house	7.4
big dog	7.2

Association measures extensions

Extensions:

$$I_1(a,b,c) = \log_2 \frac{P(abc)}{P(a)P(b)P(c)}$$

$$I_1'(a,b,c) = \log_2 \frac{P(abc)\,f(abc)}{P(a)P(b)P(c)}$$

$$H(a,b,c) = \begin{cases} 2\log_2 \dfrac{P(abc)}{P(a)P(c)}, & \text{stop(b)} \\ I_1(a,b,c), & \neg\text{stop(b)} \end{cases}$$

Evaluation of AMs

- Needed:

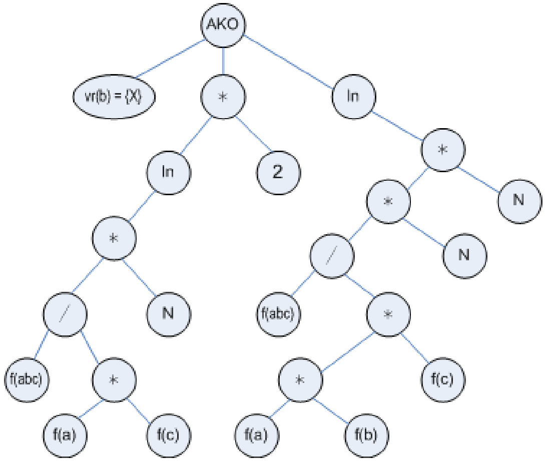
  sample of collocations and non-collocations

- F_1 measure:

$$F_1 = \frac{2 \cdot P \cdot R}{P + R}$$

Our approach based on genetic programming

- Similar to genetic algorithm
  - Population
  - Selection
    - Fittness function
    - Crossover
  - Mutation



- GP: Evolution of programs in the forms of trees

Genetic programming

- Idea - evolution of association measures

- Fitness function - F_1

$$fittness(j) = F_1(j) + \eta \frac{L_{\max} - L(j)}{L_{\max}}$$

Genetic programming

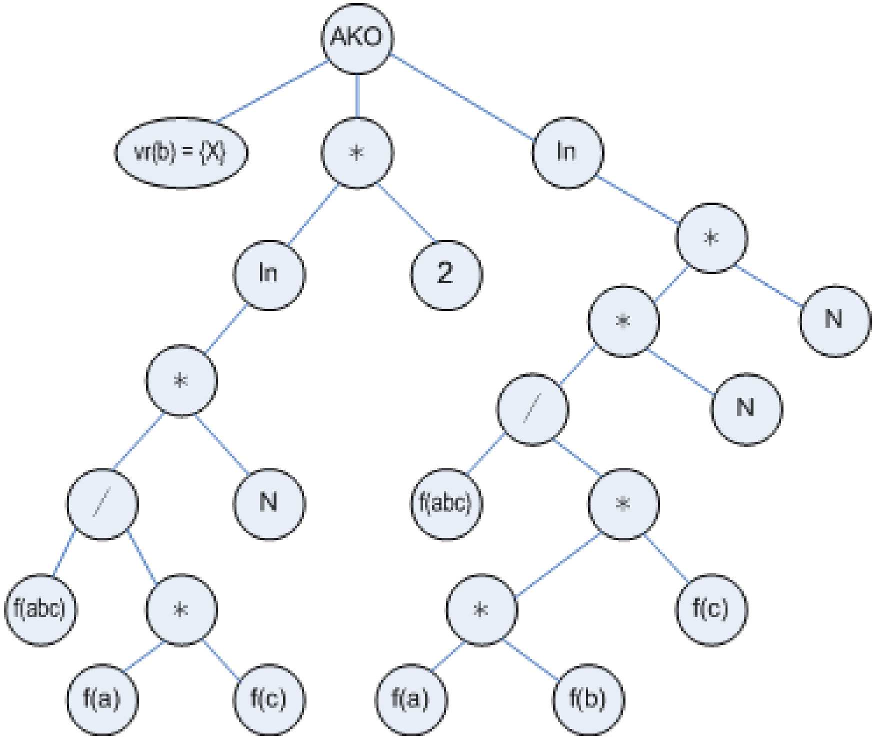
- Idea - evolution of association measures
- Fitness function - F_1

$$fitness(j) = F_1(j) + \eta \frac{L_{\max} - L(j)}{L_{\max}}$$

- Specifics:
  - Parsimony pressure
  - Stopping conditions - maximal generalisations
  - Inclusion of known AMs in the initial population

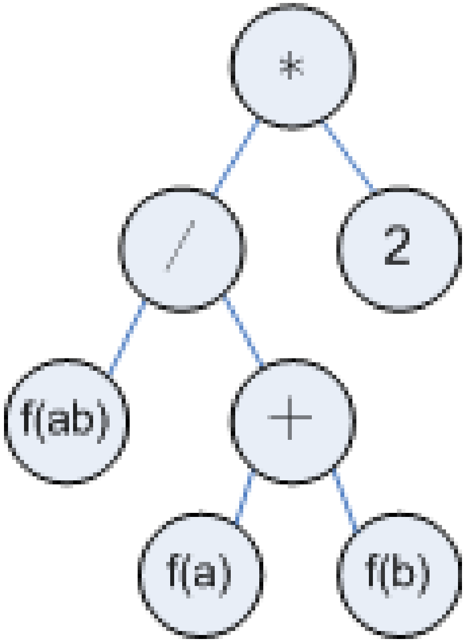
Nodes and leaves

Operators	Operands
+, -	const
*, /	f(.)
ln(|x|)	N
IF(*cond*, *a*, *b*)	POS(*W*)

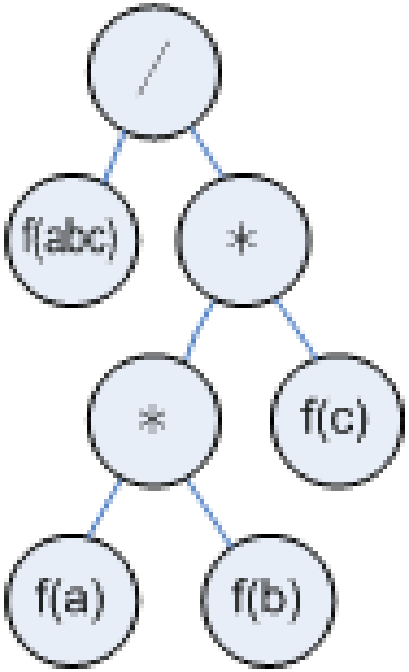


Examples

DICE coefficient:

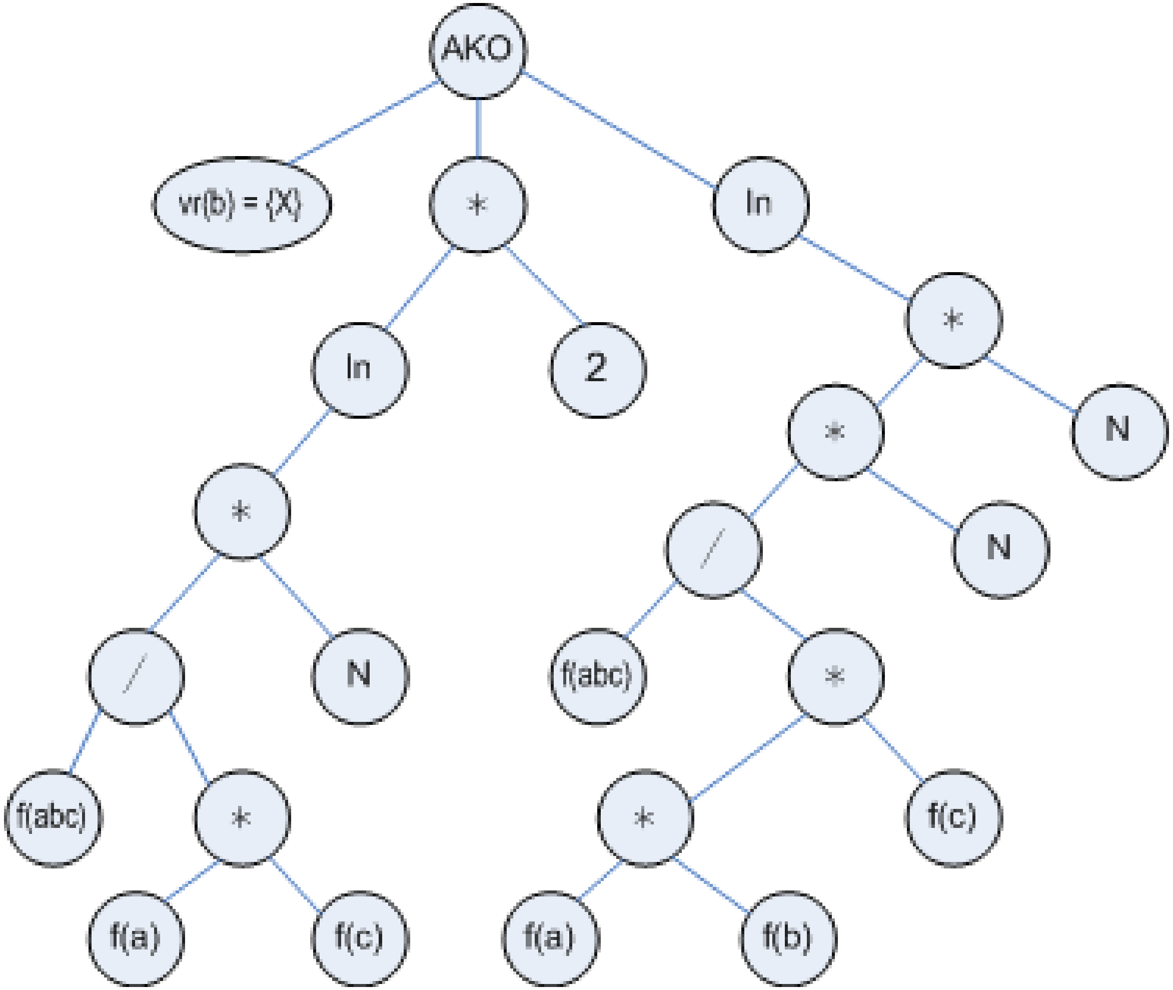


MI:



One solution

Heuristics H:

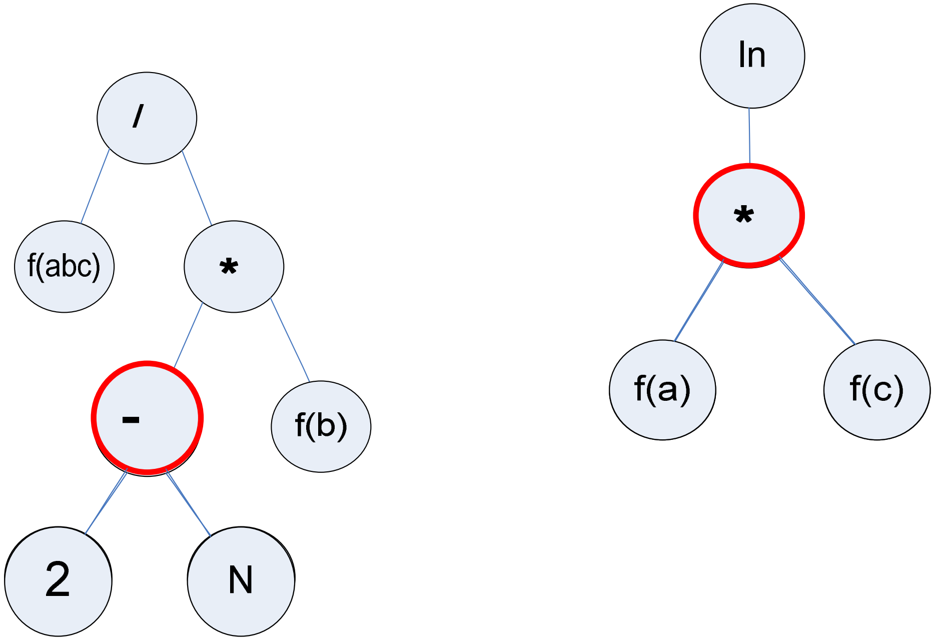


Recombination (crossover)

- Exchange of subtrees

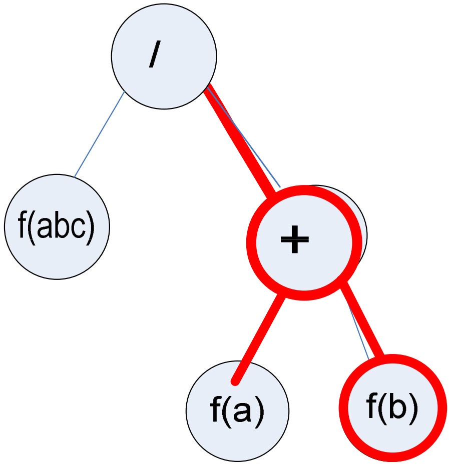
parents

children

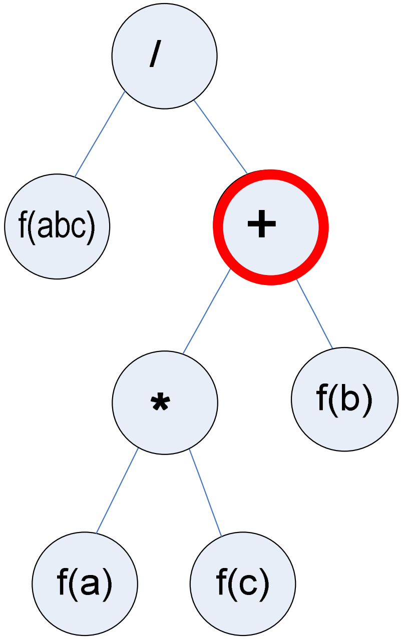


Mutation

Node insertion:



Node removal:

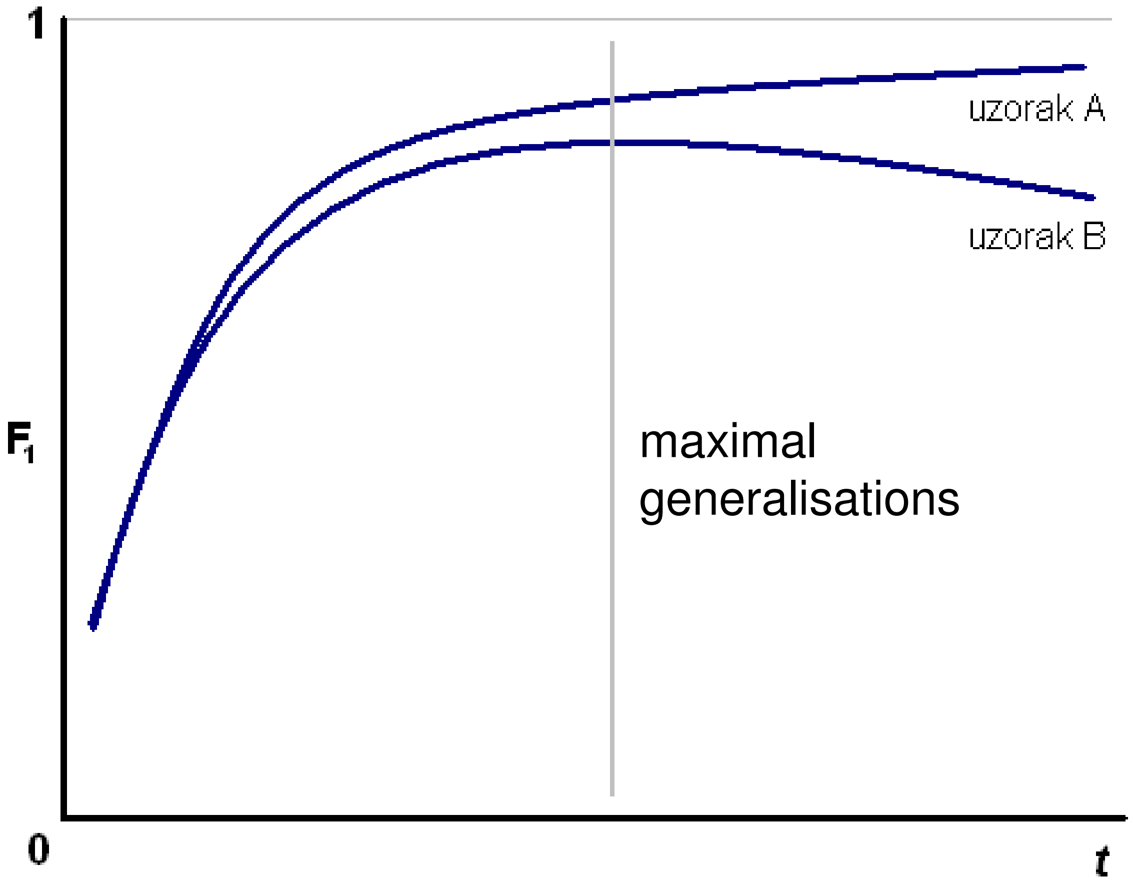


Experiment

- Collection of 7008 legislative documents
- Trigram extraction – 1.6 million
- Two samples of classified trigrams:
  - Each sample 100 positive + 100 negative examples

Generalisation

Stopping conditions – maximal generalisations



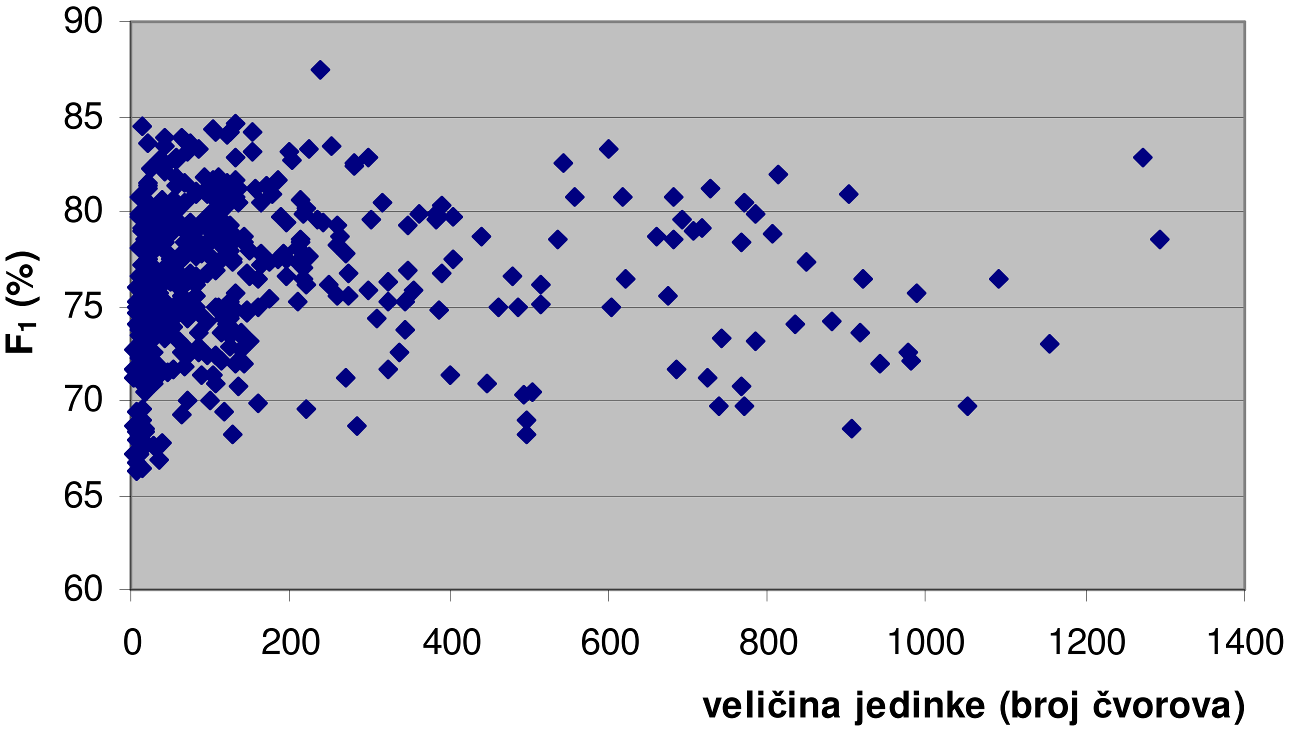
Experimental settings

- We used *three-tounament selection*
- We varied the following parameters:
  - probability of mutation [0.0001, 0.3]
  - parsimony factor [0, 0.5]
  - maximum number of nodes [20, 1000]
  - number of iterations before stopping [10^4, 10^7]
- In total, 800 runs of the algorithm (with different combinations of mentioned parameters)

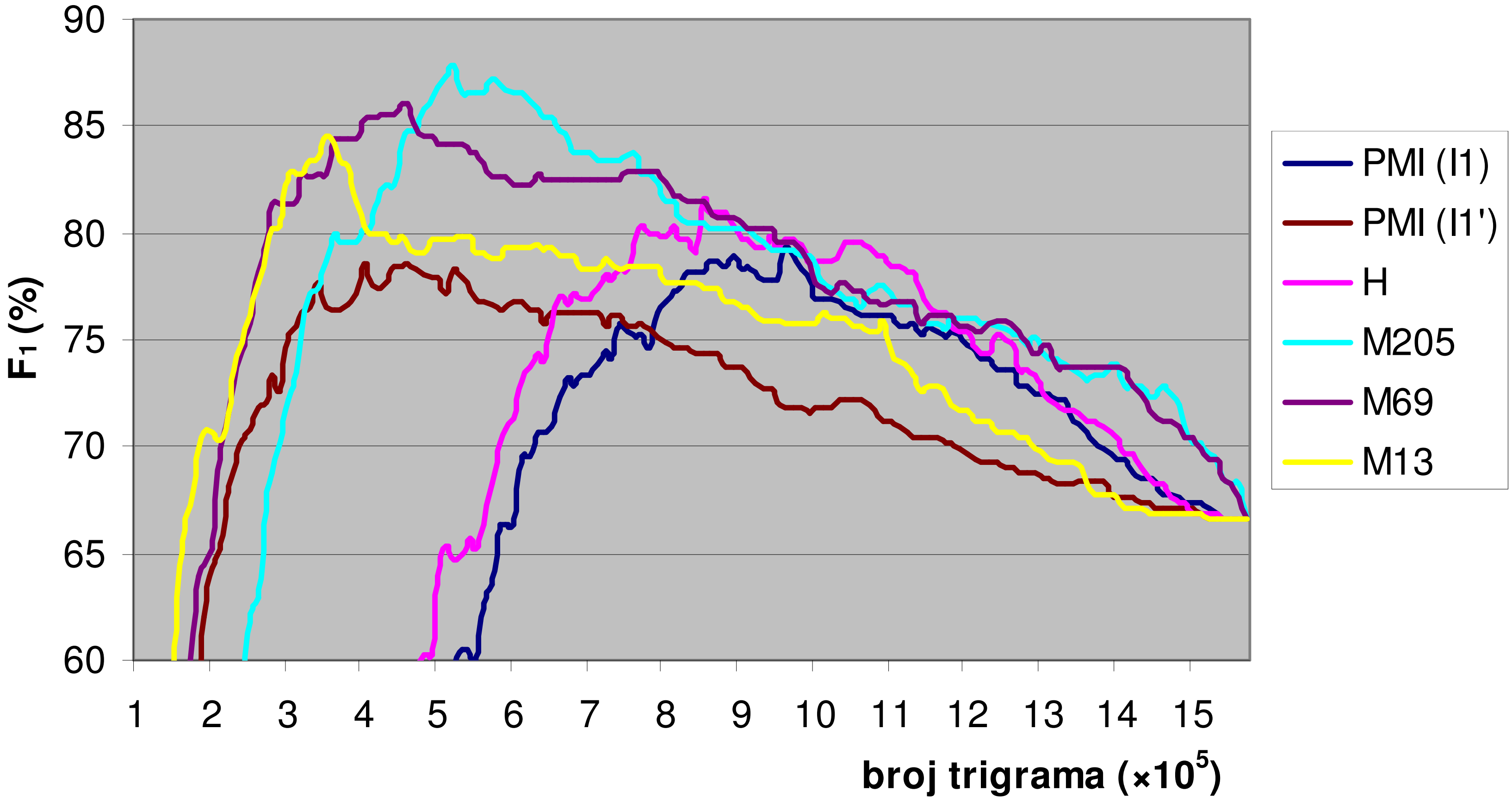
Results

- About 20% of evolved AMs reach F_1 over 80%

Figure shows F1 score and number of nodes



Results


90
85
80
75
70
65
60
F_1 (%)
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
broj trigrama ($\times 10^5$)
PMI (I1)
PMI (I1')
H
M205
M69
M13

Results

Interpretation of evolved measures in not easy (M205):

```
f(abc) f(a) f(c) * / f(abc) f(ab) f(c) - f(c) f(bc) f(b) -f(abc) + / + / N * f(b) + *
ln f(c) f(b) * * N f(a) * f(abc) f(a) f(abc) f(a) f(c) * / f(bc) * f(bc) f(b) + /
f(a) N AKO(vr(b)={X}) * (-14.426000) f(b) + / N * f(bc) f(b) -(2.000000) * ln ln /
f(a) f(c) * (2.000000) * ln ln / N * ln * / f(bc) * f(bc) f(b) + / N * (-14.426000)
f(b) + / N * f(abc) N f(a) * f(a) f(abc) f(a) f(c) * / f(bc) * f(abc) f(b) + / N * (-
14.426000) f(b) + / N * f(b) f(c) * ln ln / f(abc) f(a) f(c) * / f(c) * ln ln
(2.000000) * ln ln / N * / N * / N * ln f(c) * / f(a) f(b) + * ln ln f(abc) f(abc)
f(a) f(a) N AKO(vr(b)={X}) (-14.426000) f(b) + * / N * / N * ln f(c) * / f(a) f(b) +
* ln ln * ln ln / f(abc) f(a) f(c) * / f(a) f(b) + * ln ln (2.000000) * ln ln / N *
ln ln AKO(vr(c)={X}) N * AKO(vr(b)={X})
```

- Verification on other collections

Results

Some results are more easily interpretable (M13):

```
(-0.423000) f(c) * f(abc) / f(a) * f(abc) f(b)
- AKO(POS(b)={X}) f(abc) /
```

$$M13(a,b,c) \approx \begin{cases} \dfrac{2f(abc)^2}{f(a)f(c)}, & \text{stop(b)} \\ \dfrac{f(abc)}{f(b)}, & \neg\text{stop(b)} \end{cases}$$

Results

- 96% of measures with F_1 over 82% contain operator IF with condition “second word is a stopword”.

Conclusion

- Standard measures are imitated by evolution

- Genetic programming can be used to boost collocation extraction results for a particular corpus and to “invent” new AMs

- Futher reasearch is needed:
  - Other test collections (domains, languages)
  - Extraction of digrams, tetragrams...

Thank you