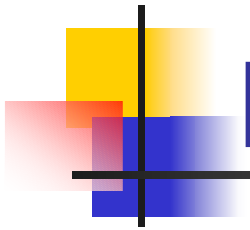
FiGO: Finding GO Terms in Unstructured Text

Francisco M. Couto,
Mário J. Silva and Pedro Coutinho

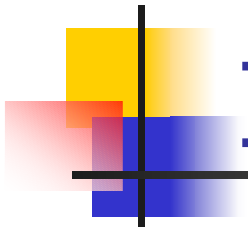
Outline

- Introduction
- Method
- Results
- Conclusions



Main Idea

- Information Content of GO terms' words
- Example: "punt binding"
  - The probability of the term being mentioned is higher if "punt" occurs than if only "binding" occurs
  - Because "binding" is also used in many other terms
  - "punt" is more informative than "binding"



Information Content

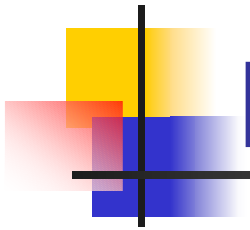
- Inversely proportional to the number of occurrences
- Information Content of an object Φ:

$$IC(\Phi) = -\log(\#\Phi/\#max)$$

  - $\#\Phi$ represents the number of times that Φ occurs
  - #max represents the maximum number of times that an object occurs
  - Log function just to improve the calculation

Outline

- Introduction
- Method
- Results
- Conclusions



FiGO

- Input:
  - A collection of terms
  - A piece of text
- Output:
  - A ranked list of terms mentioned on the piece of text

Pre-Processing (1)

- Identify all words present in the terms' names
- Ignore the stop words
  - Such as: 'in' or 'on'
- Compute the number of occurrences of each word #w
  - #w is the number of terms that have w in its name
- Compute the information content of each word:

$$IC(w)=-\log(\#w/\#max)$$

Pre-Processing (2)

- For each term's name n composed by the words: $w_0,\ldots,w_k$ compute the information content of n:

$$IC(n) = \sum IC(w_i)$$

- Compute the information content of the term's names $n_0,\ldots,n_k$:

$$IC(t) = \max\{ IC(n_i) \}$$

Example

- Considering:
  - The term t =“punt binding”
  - With the synonym “punt activity”
  - #punt=1, #binding=4, #activity=8, #max=16
- IC(“punt”)=-log(1/16)=4,
- IC(“binding”)=-log(4/16)=2
- IC(“activity”)=-log(8/16)=1
- IC(“punt binding”)=4+2=6
- IC(“punt activity”)=4+1=5
- IC(t)=max{6,5}=6

Procedure (1)

- Compute the local information content of each term t in the piece of text p:

$$LIC(t,p) = \sum IC(w_i)$$

  - Where $w_0,\ldots,w_l$ are words of the term's name that occur in p
  - $IC(t) \geq LIC(t,p)$
- LIC(t,p)/IC(t) measures how much of t's name occurs in p

Procedure (2)

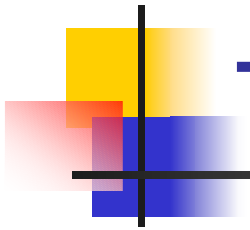
- Which terms FiGO considers mentioned?
- Each term t whose:

$$LIC(t,p) \geq \alpha \times IC(t)$$

  - $\alpha \in [0,1]$
  - When $\alpha=1$ FiGO selects only the terms fully mentioned
  - When $\alpha=0$ FiGO selects all terms

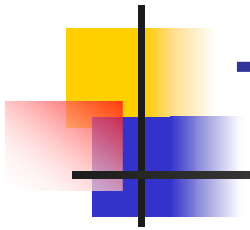
Example

- Considering:
  - The term t =“punt binding”
  - The pieces of text:
    1. “The protein has a binding activity”
    2. “The protein has a punt activity”
    3. “The protein has a punt binding activity”
  - IC(“punt”)=4 and IC(“binding”)=2
- IC(t)=6
- $LIC(t,p_1)=2$ $(\alpha \leq 1/3)$
- $LIC(t,p_2)=4$ $(\alpha \leq 2/3)$
- $LIC(t,p_3)=6$ $(\alpha \leq 1)$



Task 2.1

- Piece of text = sentence
- FiGO returned a list of sentences where the term occurred
- Which sentence?
  - Containing the protein name
  - Having the larger LIC
- If there was no sentence?
  - The most similar term (FuSSiMeG)



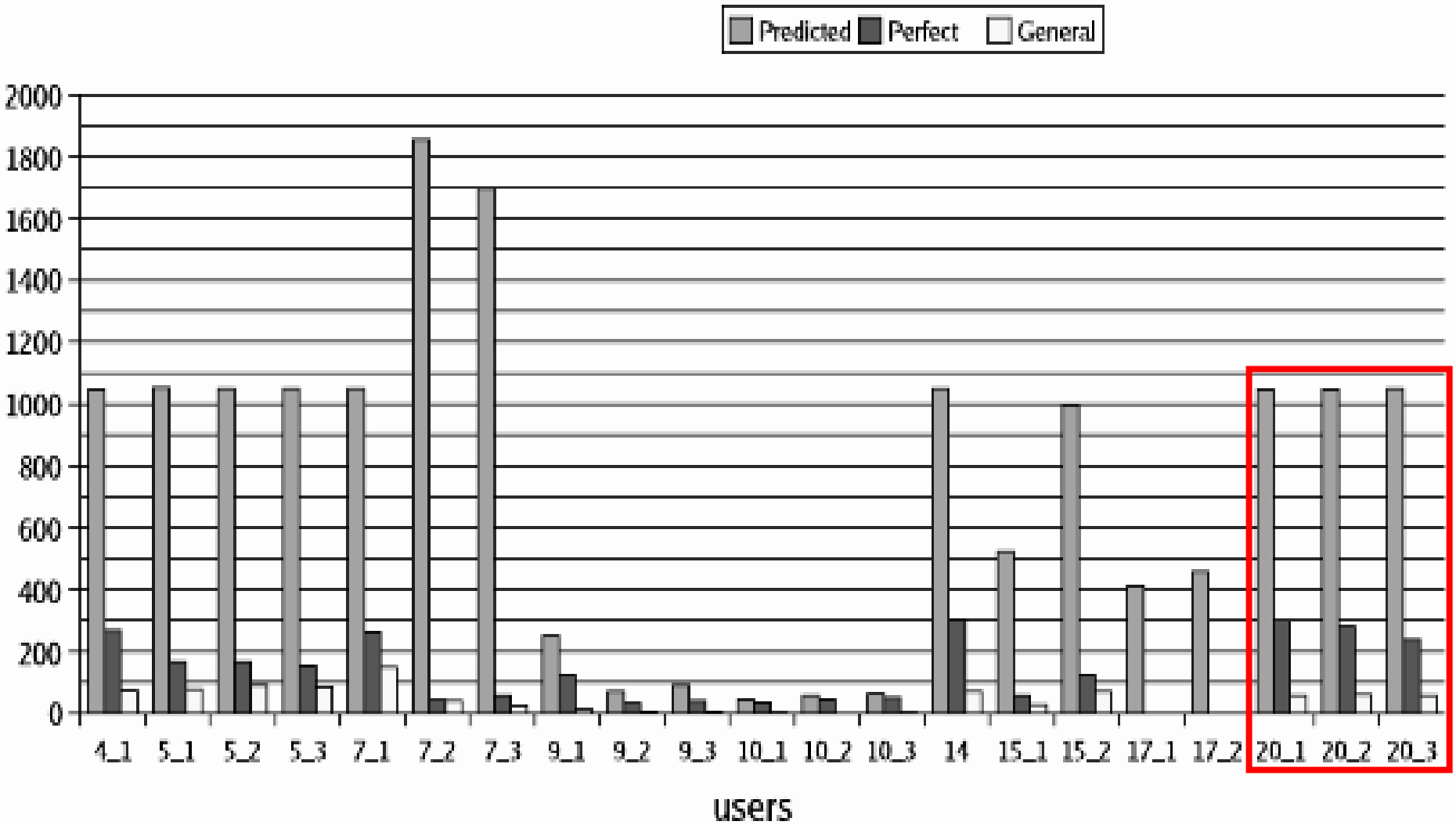
Task 2.2

- Piece of text = sentence
- FiGO returned a list of terms with the sentences where they were mentioned
- Which terms?
  - The protein name in the same sentence
  - The most meaningful annotations
    - The most infrequently annotated terms

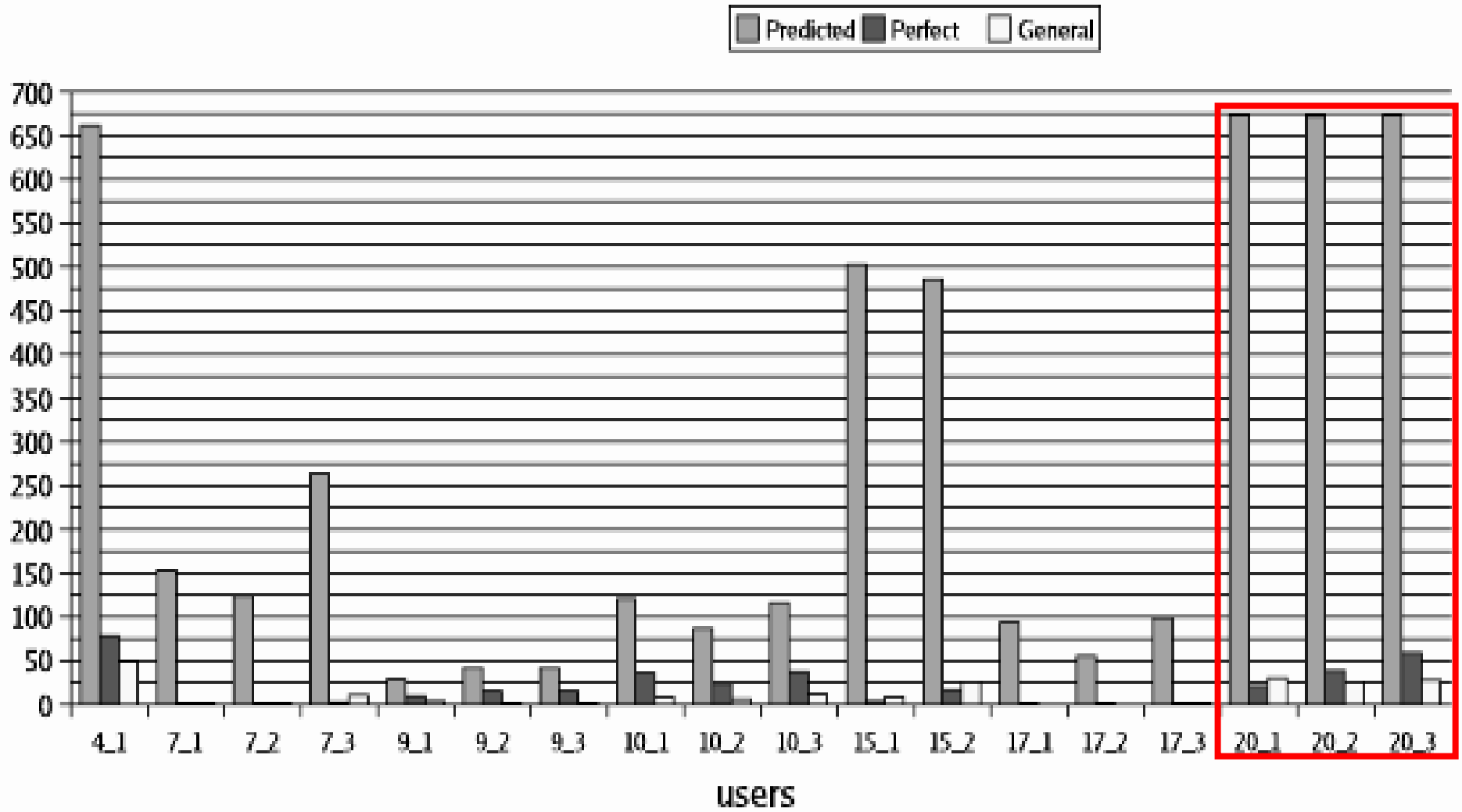
Outline

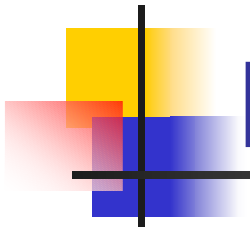
- Introduction
- Method
- Results
- Conclusions

Task 2.1 Results



Task 2.2 Results

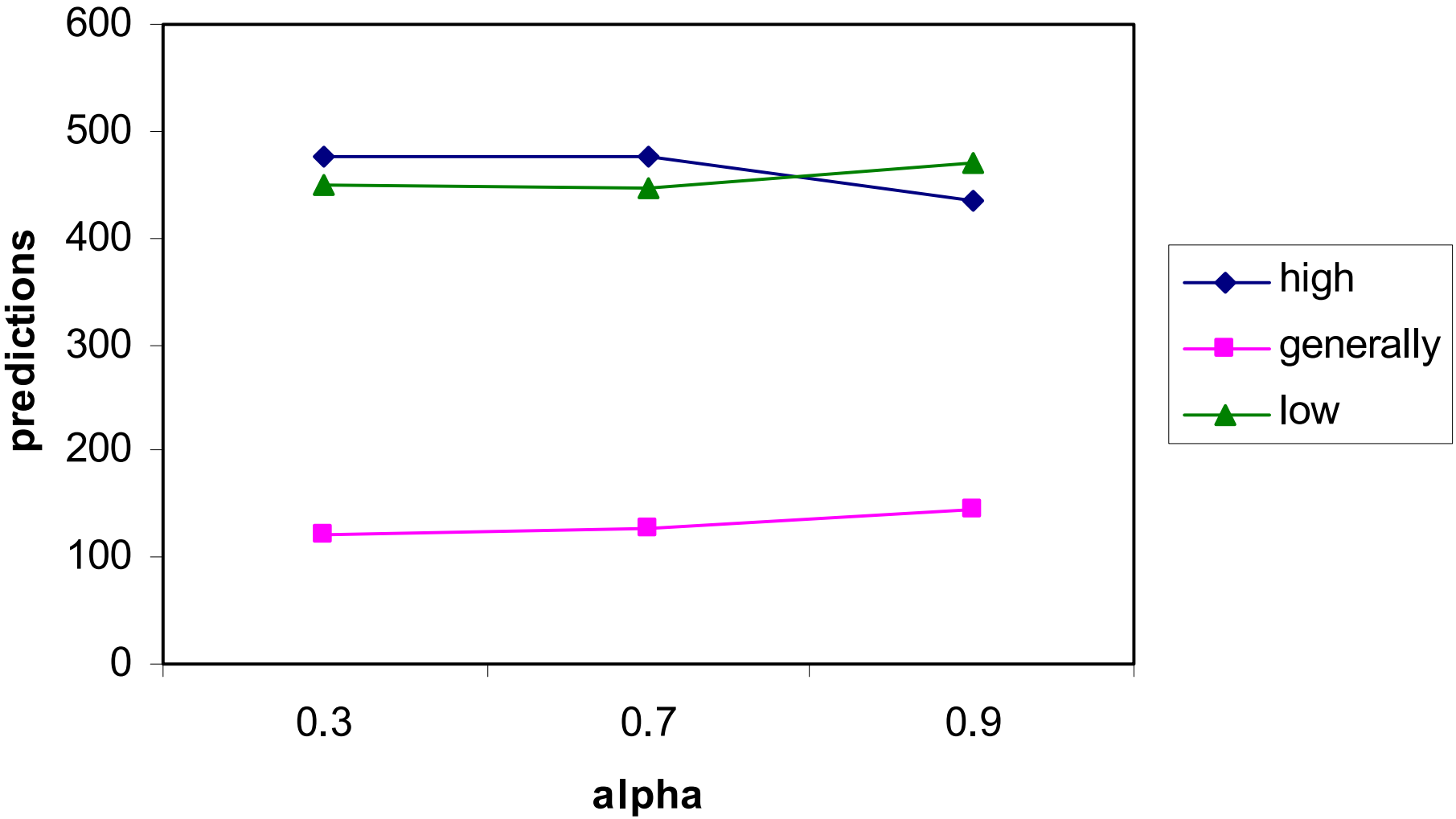




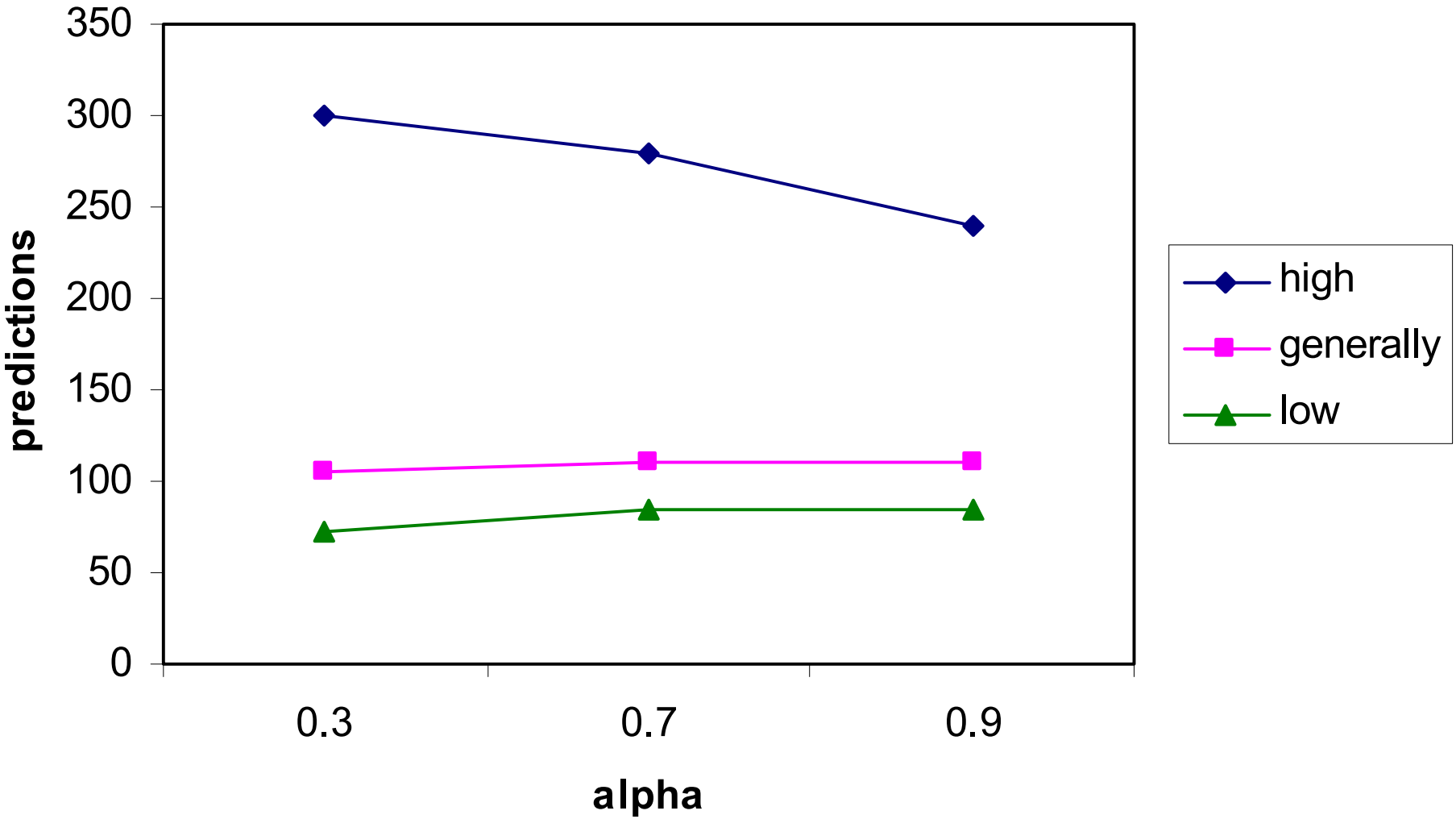
Discussion

- Very close to the largest number of perfect predictions achieved
- Our accuracy was very far from the the best results
  - Submitted the expected number of predictions
  - Some had a low confidence level
  - Filter predictions according to their confidence level to achieve better accuracy
- Better performance in task 2.1 than in task 2.2 due to greater difficulty of task 2.2

Task 2.1 GO Evaluation



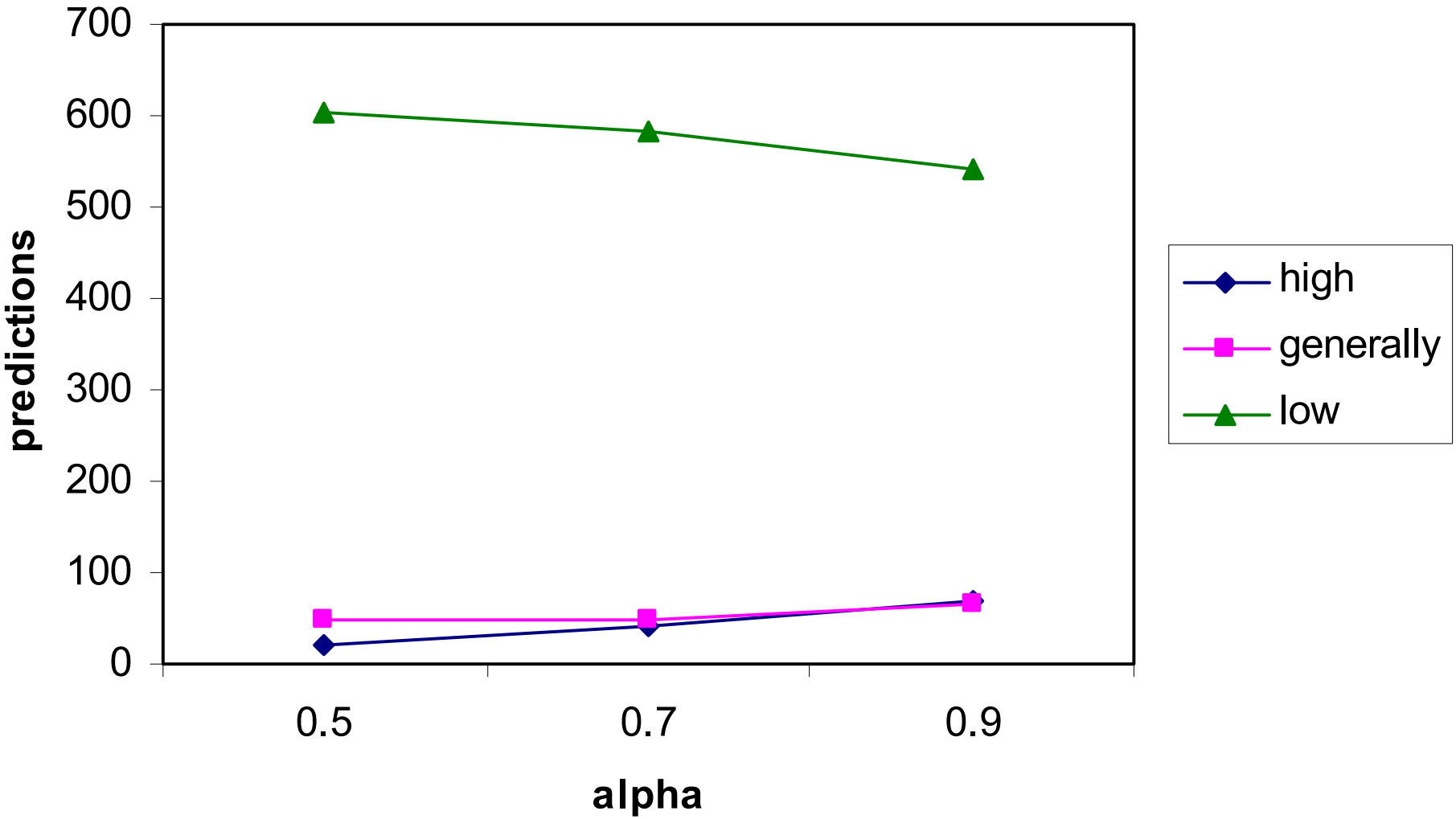
Task 2.1 Protein Evaluation



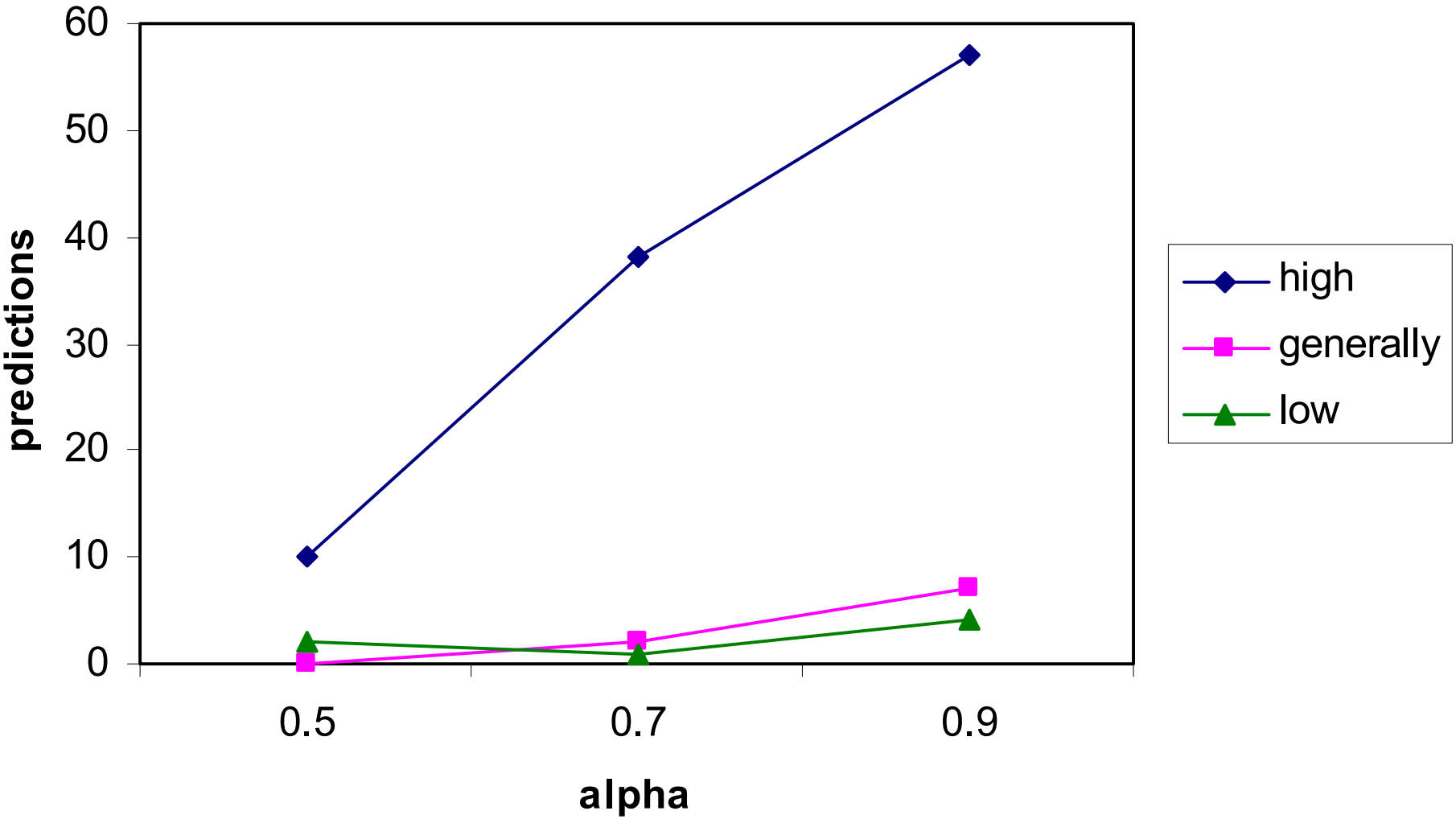
Discussion of task 2.1

- Better GO identification with a smaller α
  - Large number of terms not explicitly mentioned in the text
  - Some correct sentences had less than 70% of the GO term's name
- Better protein identification with a smaller α
  - More sentences to filter by the protein's name presence
  - About 50% of the predictions were incorrect because of the protein evaluation

Task 2.2 GO Evaluation

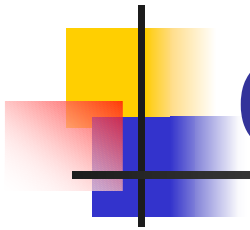


Task 2.2 Protein Evaluation



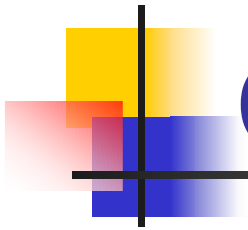
Discussion of task 2.2

- Better GO identification with a larger α
  - Terms with a larger piece of its name in the text were more accurate
  - Many terms mentioned but out of context
- Protein identification did not affect the results



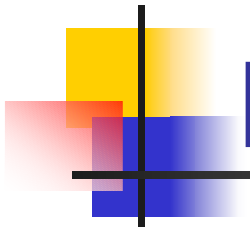
Outline

- Introduction
- Method
- Results
- Conclusions



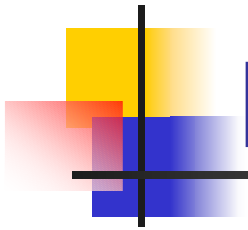
Conclusions

- FiGO a novel method for identifying GO terms in unstructured text
- Involving the information content of their names
- FiGO is fully automated, i.e. it does not need human intervention
- Despite the good score the results must be improved



Future Work

- A more effective protein identification method would likely improve our results
- The piece of text should be larger than a sentence
  - Protein and term are normally in the same paragraph but not in the same sentence
  - Number of occurrences in the document
- Task 2.2 needs domain knowledge to filter terms out of context
  - Using web resources
  - (WeBTC presented at SAC2004)



ReBIL Project

http://xldb.fc.ul.pt/rebil/