The *Large Scale Hierarchical Text Classification* PASCAL Challenge

A. Kosmopoulos[†,◇], E. Gaussier[*], G. Paliouras[†], S. Aseervatham[*]

[*] Lab. d'Informatique de Grenoble & Grenoble University, France
[†] National Center for Scientific Research "Demokritos", Greece
[◇] Athens University of Economics and Business

March 28, 2010

Outline

Introduction

Presentation of the Challenge
- Datasets and Data Preparation
- Tasks
- Evaluation Measures

Results
- Quick Overview of Approaches
- Results per Tasks
- Scalability Tests

Conclusion and Perspectives

Large scale hierarchical text classification (1)

Situation

- Problem has been addressed in the past
  - Seminal work of Yang et. al. [5] (ca. 14,000 categories) in 2003
  - Followed by extensions in 2005 ([2, 3]) to more than 100,000 categories
- Comparison of different classifiers in different settings: flat vs hierarchical
- Continuous interest in the problem (under different forms) and continuous flow of new ideas and approaches - work by Xue et al. in 2008 [4]

Large scale hierarchical text classification (2)

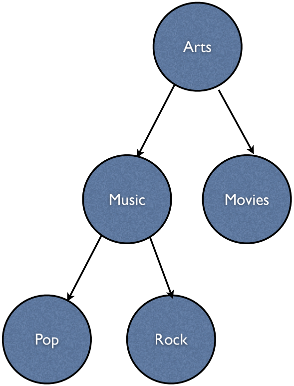
Comments

- The dichotomy introduced in early works between flat and hierarchical approaches blurred in more recent works
- Different classifiers can be used differently (e.g. feature selection or document smapling/filtering can be used in flat approaches to speed up the process); experimental space is indeed large
- Recent challenges on large scale classification on large numbers of high-dimensional exmaples, but very few categories

$\Rightarrow$ All these elements led us to propose a challenge on large scale hierarchical classification

Description of the Problem

- Hierarchy of categories is provided (flat vs hierarchical)
- Reasonable, large numbers of categories (12,000) and examples (160,000) - *don't scare potential participants with large numbers!*
- Simple hierarchical problem: documents at the leaves only
- Simple multiclass, single label problem: each document belongs to only one category



Data preparation

- Pre-processing:
  - stemming/lemmatization
  - stop-word removal
- Two type of vectors:

Content data	Description data
The Movies Net Top 20 brings you the pick of the most popular and highest-rating sites for movies on the Net today. It gives you access to the world's best movies sites, all from a single page. Movies.com features the latest movie news, reviews, trailers and a wide variety of general movie information.	Movies Top 20 - Lists links to several film sites, with brief descriptions of each.

Datasets

- Large dataset (12294 categories)
  Used for system evaluation.
- Small dataset (1139 categories)
  Used for system tuning.
- Each dataset is split into:
  - a training set (93805/4463 vectors)
  - a validation set (34905/1860 vectors)
  - a test set (34880/1858 vectors)

Tasks of the Challenge

Task Name	Content		Description	
	Train	Test	Train	Test
Task 1: Basic	✓	✓	-	-
Task 2: Cheap	-	✓	✓	-
Task 3: Expensive	✓	✓	✓	-
Task 4: Full	✓	✓	✓	✓

Tasks of the Challenge

Task Name	Train	Validation	Test
Task 1: Basic	347255	191224	194024
Task 2: Cheap	71322	39070	194024
Task 3: Expensive	368113	201487	194024
Task 4: Full	368113	201487	204288

Task Vocabulary

Evaluation Measures (1)

$$Accuracy = \frac{\text{Correct Classifications}}{D}$$

$$\text{Tree-induced error [1]} = \frac{\sum_{d=1}^{D} \text{Path-length}(c_d, t_d)}{D}$$

- D is the number of testing documents
- c_d the category in which the document was classified
- t_d the true category of the document

Evaluation Measures (2)

Macro-average precision, recall and F_1-measure

$$\text{Macro precision} = \frac{\sum_{i=1}^{M} precision_i}{M}, \; precision = \frac{TP}{TP + FP}$$

$$\text{Macro recall} = \frac{\sum_{i=1}^{M} recall_i}{M}, \; recall = \frac{TP}{TP + FN}$$

$$\text{Macro } F_1 = \frac{2 \cdot \text{Macro precision} \cdot \text{Macro recall}}{\text{Macro precision} + \text{Macro recall}}$$

M is the number of categories.

Significance Tests (1)

Comparing proportions(p-test)[6] for Accuracy

$$p = \frac{p_a + p_b}{2}$$

$$Z = \frac{p_a - p_b}{\sqrt{2p(1-p)/n}}$$

- p_a accuracy of the first system
- p_b accuracy of the second system
- n the number of trials (number of testing vectors)
- Significant different if P-value < 0.05

Significance Tests (2)

Macro sign test (S-test)[6] for Macro-average F_1-measure

$$Z = \frac{k - 0.5n}{0.5\sqrt{n}}, \text{ since } n > 12$$

- n is the number of times that a_i and b_i differ
- k is he number of times that a_i is larger than b_i
- $a_i \in [0, 1]$ is the F_1 score of system A on the ith category (i= 1, 2, ..., M)
- $b_i \in [0, 1]$ is the F_1 score of system B on the ith category (i= 1, 2, ..., M)
- M is the number of categories
- Significant different if P-value < 0.05

Standard Approaches

Two main approaches[4]:

Big-bang Directly categorize documents to the leaves.

Top-down Hierarchy is exploited in order to divide the problem into smaller ones.

Big-bang approaches are usually more accurate while Top-down approaches are usually faster.

Approaches used in the Challenge (1)

- Most participants either used a big-bang approach or exploited only a small part of the hierarchy.
- Feature selection was tried but did not always help.
- Regarding the classifiers:
  - Lazy learners were used, which are very fast at training but slower at classification. (i.e. kNN)
  - Eager learners were also used and were faster at classification. (i.e. Naïve Bayes, SVMs)

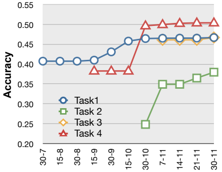
Approaches used in the Challenge (2)

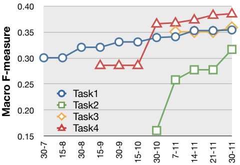
- **arthur_general**- two-level hierarchy, multi-class SVM.
- **logicators** - subset of the hierarchy, hierarchical SVMs.
- **Turing** - knn for a subset and then Naïve Bayes classifier.
- **XipengQiu** - centroid for each class and IDF of terms.
- **jhuang** - flat hierarchy, two online algorithms (OOZ and PA).
- **NakaCristo** - flat hierarchy, kNN with three variants.
- **Brouard** - IDF feature selection, linking terms to categories.
- **alpaca** - classifier combination, 2-degree polynomial SVMs.

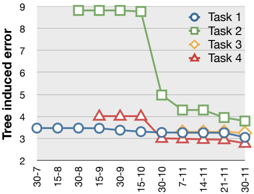
Highest Scores per Task

	Task 1	Task 2	Task 3	Task 4
Accuracy	0.467632	0.380619	0.467861	0.504759
Macro F-measure	0.35494	0.317133	0.359557	0.386195
Tree Induced Error	3.07858	3.80803	3.2621	2.82101

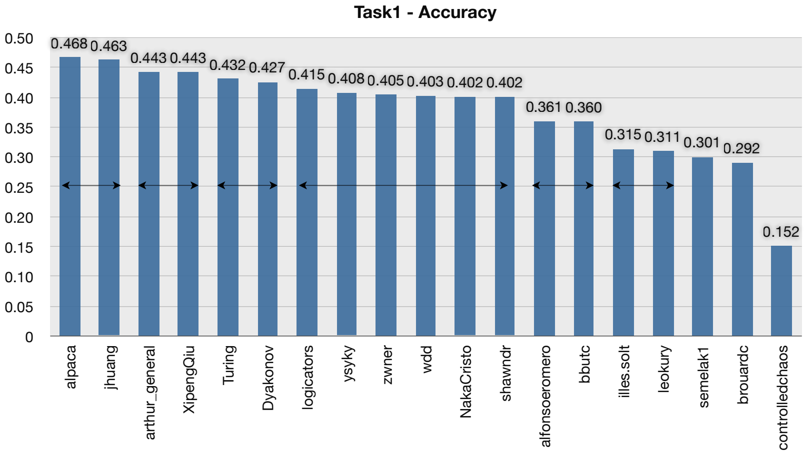
Variation of Highest Scores



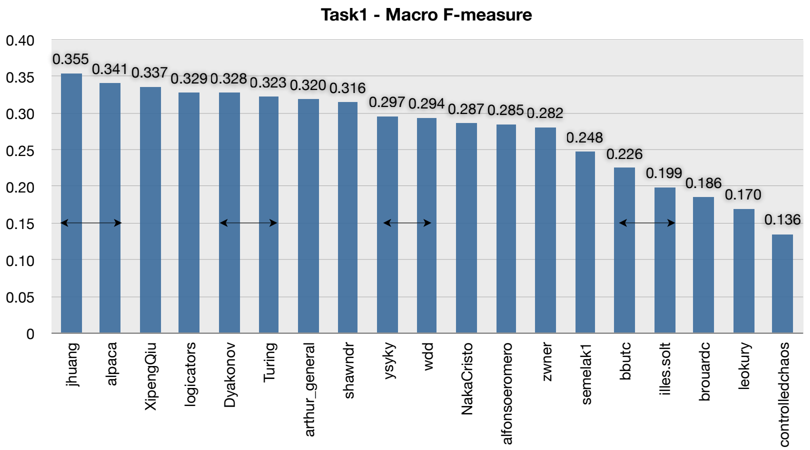




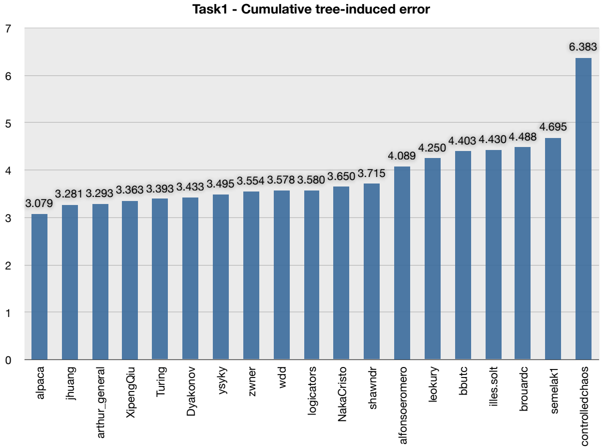
Task 1: Basic


Task1 - Accuracy
0.50
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
0.468
0.463
0.443
0.443
0.432
0.427
0.415
0.408
0.405
0.403
0.402
0.402
0.361
0.360
0.315
0.311
0.301
0.292
0.152
alpaca
jhuang
arthur_general
XipengQiu
Turing
Dyakonov
logicators
ysyky
zwner
wdd
NakaCristo
shawndr
alfonsoeromero
bbutc
illes.solt
leokury
semelak1
brouardc
controlledchaos

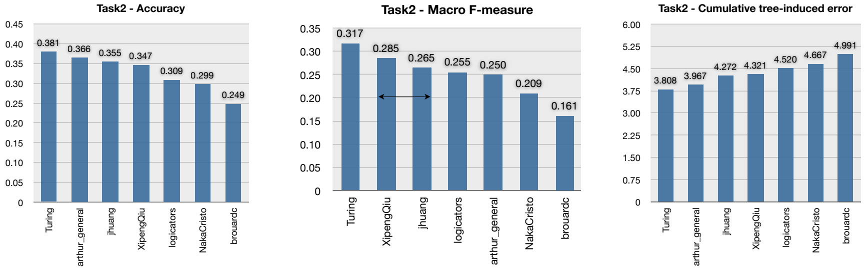
Task 1: Basic


Task1 - Macro F-measure
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
0.355
0.341
0.337
0.329
0.328
0.323
0.320
0.316
0.297
0.294
0.287
0.285
0.282
0.248
0.226
0.199
0.186
0.170
0.136
jhuang
alpaca
XipengQiu
logicators
Dyakonov
Turing
arthur_general
shawndr
ysyky
wdd
NakaCristo
alfonsoeromero
zwner
semelak1
bbutc
illes.solt
brouardc
leokury
controlledchaos

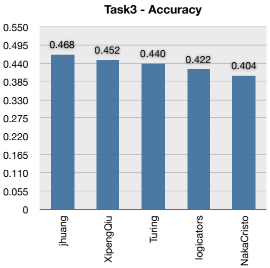
Task 1: Basic

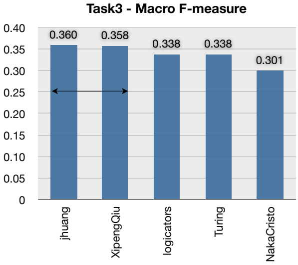

Task1 - Cumulative tree-induced error
7
6
5
4
3
2
1
0
3.079
3.281
3.293
3.363
3.393
3.433
3.495
3.554
3.578
3.580
3.650
3.715
4.089
4.250
4.403
4.430
4.488
4.695
6.383
alpaca
jhuang
arthur_general
XipengQiu
Turing
Dyakonov
ysyky
zwner
wdd
logicators
NakaCristo
shawndr
alfonsoeromero
leokury
bbutc
illes.solt
brouardc
semelak1
controlledchaos

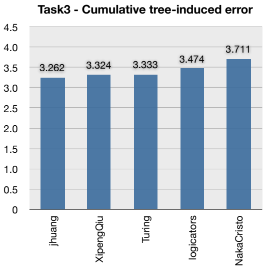
Task 2: Cheap


Task2 - Accuracy
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
0.381
0.366
0.355
0.347
0.309
0.299
0.249
Turing
arthur_general
jhuang
XipengQiu
logicators
NakaCristo
brouardc
Task2 - Macro F-measure
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
0.317
0.285
0.265
0.255
0.250
0.209
0.161
Turing
XipengQiu
jhuang
logicators
arthur_general
NakaCristo
brouardc
Task2 - Cumulative tree-induced error
6.00
5.25
4.50
3.75
3.00
2.25
1.50
0.75
0
3.808
3.967
4.272
4.321
4.520
4.667
4.991
Turing
arthur_general
jhuang
XipengQiu
logicators
NakaCristo
brouardc

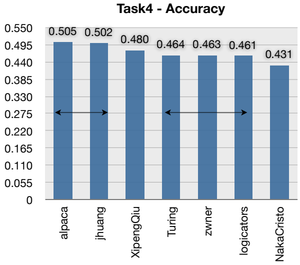
Task 3: Expensive

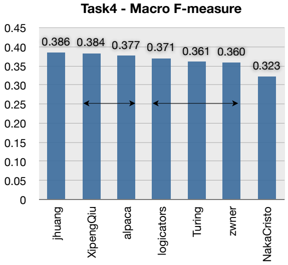


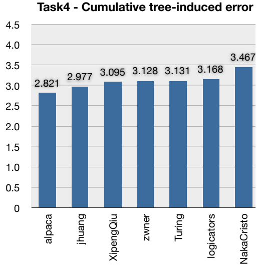




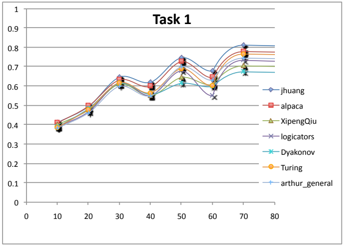
Task 4: Full

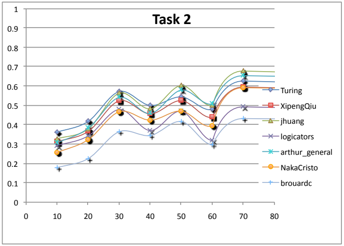

Task4 - Accuracy
0.550
0.495
0.440
0.385
0.330
0.275
0.220
0.165
0.110
0.055
0
0.505
0.502
0.480
0.464
0.463
0.461
0.431
alpaca
jhuang
XipengQiu
Turing
zwner
logicators
NakaCristo

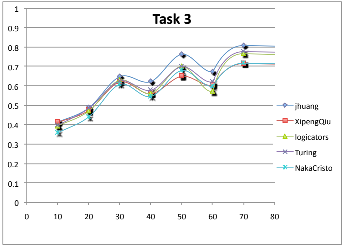

Task4 - Macro F-measure
0.45
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
0.386
0.384
0.377
0.371
0.361
0.360
0.323
jhuang
XipengQiu
alpaca
logicators
Turing
zwner
NakaCristo

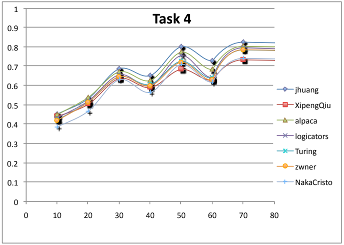

Task4 - Cumulative tree-induced error
4.5
4.0
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0
2.821
2.977
3.095
3.128
3.131
3.168
3.467
alpaca
jhuang
XipengQiu
zwner
Turing
logicators
NakaCristo

F_1-measure per Category Size









Scalability Tests - Time

Categories	XipengQiu	logicators	Turing	NakaCristo
12294	1m 12.5s 94m 8.2s	**67m 1.4s** **296m 45s**	**0m 13s** **1258m 2s**	18.8s 42m 5s
10000	0m 58s 60m 3s	41m 20.7s 153m 20s	0m 8s 779m 35s	4m 11.5s 27m 6s
1000	0m 7.6s 0m 42s	0m 35s 1m 28s	0m 0.7s 9m 36s	0m 24.2s 0m 30.2s
100	0m 4.7s 0m 1.2s	0m 0.2s 0m 3.7s	0m 13s 0m 22.6s	0m 2.5s 0m 1.9s

First line = train
Second line = classify

Scalability Tests - Memory

Categories	XipengQiu	logicators	Turing	NakaCristo
12294	2920 Mb 1382 Mb	**5700 Mb** **3900 Mb**	**200 Mb** **6000 Mb**	921 Mb 996 Mb
10000	2400 Mb 1050 Mb	4600 Mb 2950 Mb	76 Mb 5800 Mb	828 Mb 762 Mb
1000	170 Mb 149 Mb	444 Mb 434 Mb	$<$ 50 Mb 1320 Mb	110Mb 79 Mb

First line = train
Second line = classify

Conclusion and Perspectives

- All the approaches we are aware of on large scale classification tried by participants; we thus believe the results obtained represent the state-of-the-art on this collection
- No complete hierarchical approaches, a la pachinko; rather approaches with a limited use of the hierarchy
- Best results are provided by both hierarchical and flat methods; hierarchical methods seem faster
- Useful benchmark for future use; oracle will be available for a while at the challenge site
- LSHTC-2 - What should be different? Multilabel? Original text? Other data?

Ofer Dekel, Joseph Keshet, and Yoram Singer.
Large margin hierarchical classification.
In *ICML '04: Proceedings of the twenty-first international conference on Machine learning*, page 27, New York, NY, USA, 2004. ACM.

Tie-Yan Liu, Yiming Yang, Hao Wan, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma.
Support vector machines classification with a very large-scale taxonomy.
SIGKDD Explorations, 7(1):36–43, 2005.

Tie-Yan Liu, Yiming Yang, Hao Wan, Qian Zhou, Bin Gao, Hua-Jun Zeng, Zheng Chen, and Wei-Ying Ma.
An experimental study on large-scale web categorization.
In Allan Ellis and Tatsuya Hagino, editors, *WWW (Special interest tracks and posters)*, pages 1106–1107. ACM, 2005.

Gui-Rong Xue, Dikan Xing, Qiang Yang, and Yong Yu.
Deep classification in large-scale text hierarchies.
In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 619–626, New York, NY, USA, 2008. ACM.

B. Kisiel Y. Yang, J. Zhang.
A scalability analysis of classifiers in text.
In *ACM SIGIR Conference*. ACM, 2003.

Yiming Yang and Xin Liu.
A re-examination of text categorization methods.
pages 42–49. ACM Press, 1999.