Building Automated Text Classifiers via Machine Learning

Fabrizio Sebastiani

Istituto di Scienza e Tecnologie dell'Informazione
Consiglio Nazionale delle Ricerche
56124 Pisa, Italy
E-mail: fabrizio.sebastiani@isti.cnr.it

ISKO UK / BCS IRSG Joint Workshop

Outline

1. Introduction to ATC
2. The Machine Learning approach to ATC
3. Effectiveness and Efficiency
4. What next?

Outline

1 Introduction to ATC

2 The Machine Learning approach to ATC

3 Effectiveness and Efficiency

4 What next?

Introduction to ATC

- (Automated) Text Classification (ATC): building software systems that assign classes from a predefined classification scheme to textual documents [9, 10]
- Discipline at the crossroads of several fields in computer science, including
  - machine learning
  - information retrieval
  - computational linguistics / natural language processing
- Synonyms:
  - Text Classification → text categorization, text coding, document classification, ...
  - Classification Scheme → category set, codeframe, codebook, controlled vocabulary, ...
  - Classes → categories, codes, index terms, ...

Introduction to ATC (cont'd)

Several variants of ATC have been addressed in the scientific literature:

- according to cardinality:
  - binary or multi-class ("how many classes are there in the classification scheme?")
  - single-label or multi-label ("how many classes may be attributed to the same text?")
- according to structure of the classification scheme:
  - flat or hierarchical,
  - nominal or ordinal ("is there a linear order defined on the classification scheme?")
  - universal or specific;
- according to dimension:
  - by topic
  - by sentiment
  - by genre
  - by author

Introduction to ATC (cont'd)

Several variants of ATC have been addressed in the scientific literature:

- according to cardinality:
  - binary or multi-class ("how many classes are there in the classification scheme?")
  - single-label or multi-label ("how many classes may be attributed to the same text?")
- according to structure of the classification scheme:
  - flat or hierarchical,
  - nominal or ordinal ("is there a linear order defined on the classification scheme?")
  - universal or specific;
- according to dimension:
  - by topic
  - by sentiment
  - by genre
  - by author

Introduction to ATC (cont'd)

Several variants of ATC have been addressed in the scientific literature:

- according to cardinality:
  - binary or multi-class ("how many classes are there in the classification scheme?")
  - single-label or multi-label ("how many classes may be attributed to the same text?")
- according to structure of the classification scheme:
  - flat or hierarchical,
  - nominal or ordinal ("is there a linear order defined on the classification scheme?")
  - universal or specific;
- according to dimension:
  - by topic
  - by sentiment
  - by genre
  - by author

Outline

1 Introduction to ATC

2 The Machine Learning approach to ATC

3 Effectiveness and Efficiency

4 What next?

The Machine Learning approach to ATC

- Early '90s: switch from the "knowledge engineering" to the "machine learning" approach to text classification;
- The learning metaphor: the system learns from a sample of manually classified texts (the training set) the characteristics a new text should have in order to be attributed a given class;
- The training set needs to include positive examples of the class and negative examples of the class;
- Training works by detecting linguistic patterns (e.g., words, word *n*-grams, noun phrases, syntactic patterns, etc.) that are distributed most differently in the positive and in the negative training examples;
- Providing manually classified examples of the class to the system is by no means different than providing a child with (positive and negative) examples of, say, what a tiger is, in order to teach him to recognize tigers.

The Machine Learning approach to ATC

- Early '90s: switch from the "knowledge engineering" to the "machine learning" approach to text classification;
- The learning metaphor: the system learns from a sample of manually classified texts (the training set) the characteristics a new text should have in order to be attributed a given class;
- The training set needs to include positive examples of the class and negative examples of the class;
- Training works by detecting linguistic patterns (e.g., words, word *n*-grams, noun phrases, syntactic patterns, etc.) that are distributed most differently in the positive and in the negative training examples;
- Providing manually classified examples of the class to the system is by no means different than providing a child with (positive and negative) examples of, say, what a tiger is, in order to teach him to recognize tigers.

The Machine Learning approach to ATC

- Early '90s: switch from the "knowledge engineering" to the "machine learning" approach to text classification;
- The learning metaphor: the system learns from a sample of manually classified texts (the training set) the characteristics a new text should have in order to be attributed a given class;
- The training set needs to include positive examples of the class and negative examples of the class;
- Training works by detecting linguistic patterns (e.g., words, word *n*-grams, noun phrases, syntactic patterns, etc.) that are distributed most differently in the positive and in the negative training examples;
- Providing manually classified examples of the class to the system is by no means different than providing a child with (positive and negative) examples of, say, what a tiger is, in order to teach him to recognize tigers.

This is a tiger!

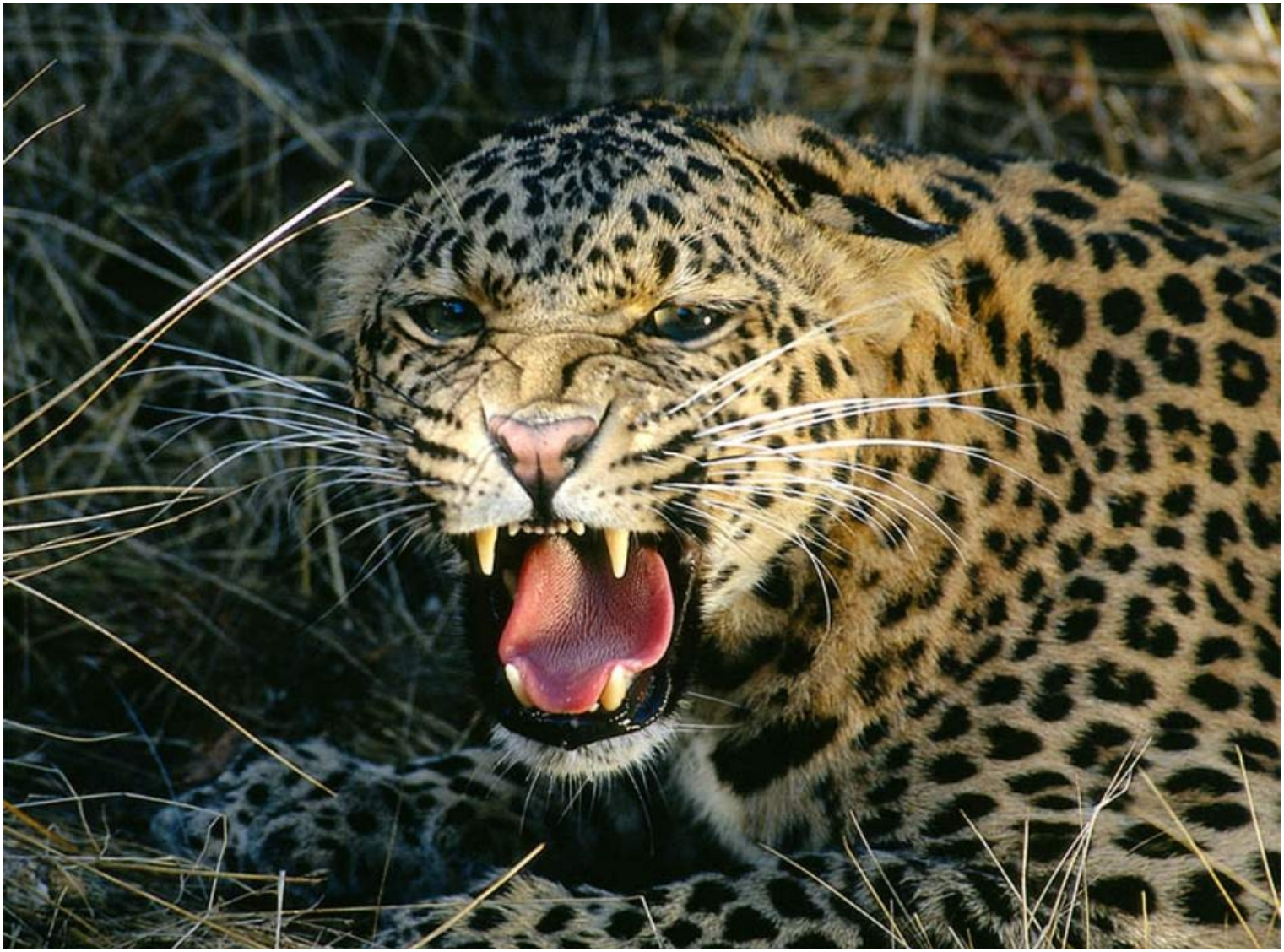
This is another tiger!

This is yet another tiger!

Also a tiger!

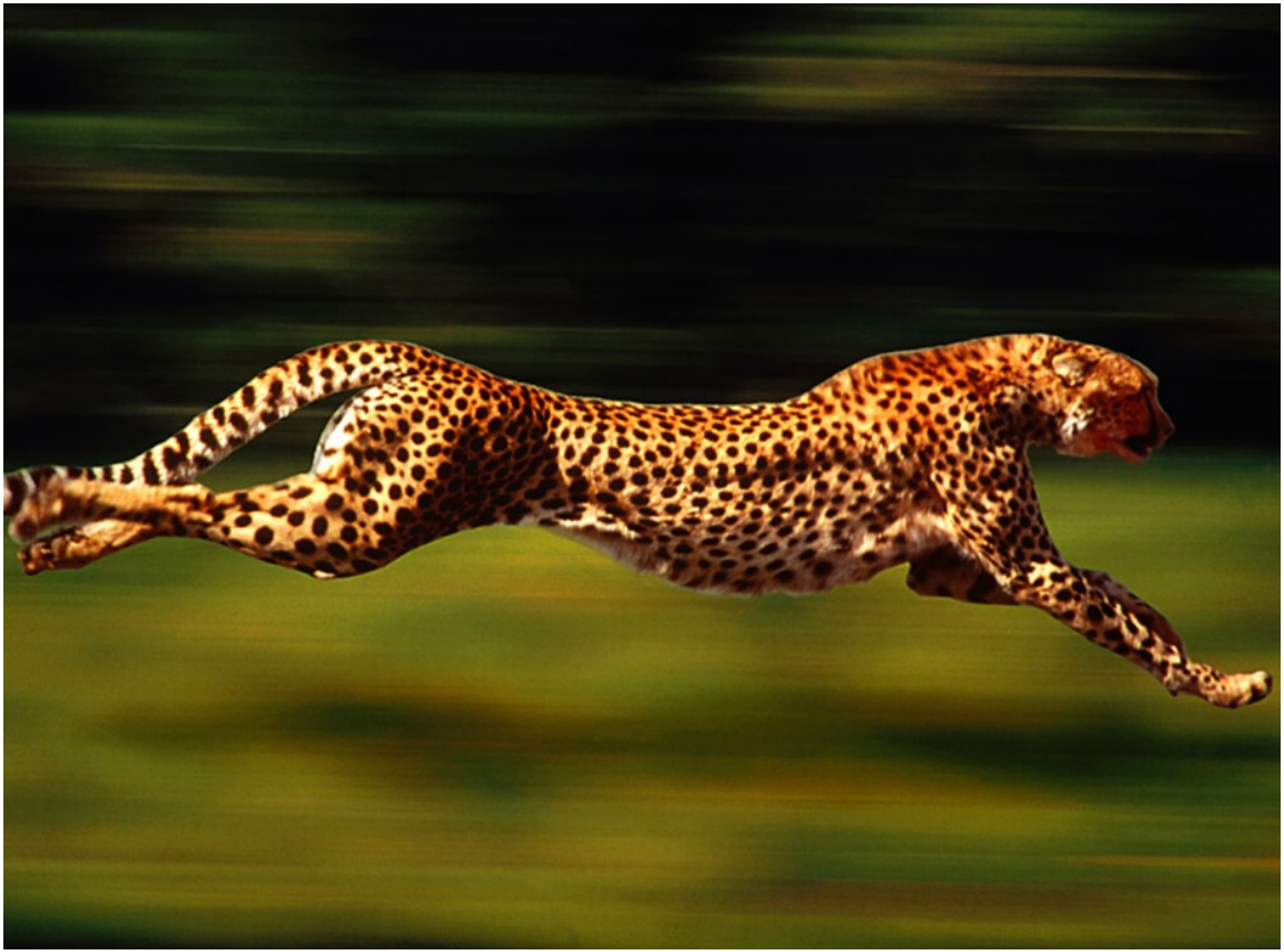
This is a NOT a tiger!

NOT a tiger either!



Absolutely NOT a tiger!

Is this a tiger?



Advantages of the machine learning approach

- No need to manually write classification rules; the system only needs manually classified examples for training, which in some cases may already be available;
- Easy update to
  - revised classification scheme
  - brand new classification scheme

  since the system only needs user-classified examples for training that reflect the new situation;
- No need of specialized / domain-dependent dictionaries;
- State-of-the-art accuracy, excellent learning and classification speed;
- Possibility to provide feedback to the system, thus allowing it to implement continuous learning.

The ATC Ecosystem

1. The Classifier Training module
   - Allows to train classifiers, given a classification scheme and a training set of documents
2. The Accuracy Estimation module
   - Estimates the likely accuracy of the trained classifiers
3. The Classification module
   - Allows to classify yet unclassified data by using the trained classifiers
4. The Validation module
   - Allows the user to manually check the most uncertain class assignments
5. The Training Data Cleaning module
   - Allows the user to check the training documents most likely to be misclassified
6. The Proactive Learning module
   - Allows the user to manually check the automatically classified documents most beneficial to retraining

The ATC Ecosystem

1. The Classifier Training module
   - Allows to train classifiers, given a classification scheme and a training set of documents
2. The Accuracy Estimation module
   - Estimates the likely accuracy of the trained classifiers
3. The Classification module
   - Allows to classify yet unclassified data by using the trained classifiers
4. The Validation module
   - Allows the user to manually check the most uncertain class assignments
5. The Training Data Cleaning module
   - Allows the user to check the training documents most likely to be misclassified
6. The Proactive Learning module
   - Allows the user to manually check the automatically classified documents most beneficial to retraining

The ATC Ecosystem

1. The Classifier Training module
   - Allows to train classifiers, given a classification scheme and a training set of documents
2. The Accuracy Estimation module
   - Estimates the likely accuracy of the trained classifiers
3. The Classification module
   - Allows to classify yet unclassified data by using the trained classifiers
4. The Validation module
   - Allows the user to manually check the most uncertain class assignments
5. The Training Data Cleaning module
   - Allows the user to check the training documents most likely to be misclassified
6. The Proactive Learning module
   - Allows the user to manually check the automatically classified documents most beneficial to retraining

The ATC Ecosystem

1. The Classifier Training module
   - Allows to train classifiers, given a classification scheme and a training set of documents
2. The Accuracy Estimation module
   - Estimates the likely accuracy of the trained classifiers
3. The Classification module
   - Allows to classify yet unclassified data by using the trained classifiers
4. The Validation module
   - Allows the user to manually check the most uncertain class assignments
5. The Training Data Cleaning module
   - Allows the user to check the training documents most likely to be misclassified
6. The Proactive Learning module
   - Allows the user to manually check the automatically classified documents most beneficial to retraining

The ATC Ecosystem

1. The Classifier Training module
   - Allows to train classifiers, given a classification scheme and a training set of documents
2. The Accuracy Estimation module
   - Estimates the likely accuracy of the trained classifiers
3. The Classification module
   - Allows to classify yet unclassified data by using the trained classifiers
4. The Validation module
   - Allows the user to manually check the most uncertain class assignments
5. The Training Data Cleaning module
   - Allows the user to check the training documents most likely to be misclassified
6. The Proactive Learning module
   - Allows the user to manually check the automatically classified documents most beneficial to retraining

The ATC Ecosystem

1. The Classifier Training module
   - Allows to train classifiers, given a classification scheme and a training set of documents
2. The Accuracy Estimation module
   - Estimates the likely accuracy of the trained classifiers
3. The Classification module
   - Allows to classify yet unclassified data by using the trained classifiers
4. The Validation module
   - Allows the user to manually check the most uncertain class assignments
5. The Training Data Cleaning module
   - Allows the user to check the training documents most likely to be misclassified
6. The Proactive Learning module
   - Allows the user to manually check the automatically classified documents most beneficial to retraining

Outline

1 Introduction to ATC

2 The Machine Learning approach to ATC

3 Effectiveness and Efficiency

4 What next?

Effectiveness and Efficiency

- Scientific research on ATC thoroughly investigates issues of effectiveness (i.e., classification accuracy) and efficiency (i.e., training and classification speed) via controlled experiments on publicly available datasets of manually classified data;
- Often used datasets, ranging from small to very large:

Dataset	Type of documents	# of classes	# of training docs	# of test docs
Reuters-21578	Newswire	115	9,604	3,299
WIPO-Alpha	Patents	614	46,324	28,926
OHSUMED	Scientific	97	183,229	50,216
Yahoo! Directory	Web	132,199	492,617	275,364
RCV1-v2	Newswire	101	23,149	781,265

Effectiveness Issues

- Effectiveness is tested by training a classifier on a set of training documents, applying it to the test documents, and checking the degree of correspondence between the results and the original, manually attributed classes.
- Standard mathematical measures have been defined (e.g., F_1) that reward both
  - the ability of the classifier to avoid false positives (precision)
  - the ability of the classifier to avoid false negatives (recall)
- Today's technology allows to obtain, e.g., on Reuters-21578,
  - $F_1 \geq 0.87$, as an average across all 115 classes (from 1 to 3,200+ training examples per class);
  - $F_1 \geq 0.93$, as an average across the 10 most frequent classes (from 300 to 3,200+ training examples per class).
- Accuracy depends on many factors, including number of positive training documents per class, average document length, and intrinsic class "difficulty".

Efficiency Issues

- Efficiency is tested by recording training times and classification times on a given dataset.
- Classification time is usually regarded as more important than training time, since
  - training can be performed off-line;
  - training is performed once for all.
- Both training times and classification times are usually not a problem, given today's hardware; e.g., on our systems
  - training on 1,000 documents for a 20-class classification scheme takes less than 2 mins;
  - classifying 100,000 documents for a 20-class classification scheme takes less than 8 mins.
- Both training and testing are much quicker when the classification scheme is organized hierarchically.

Outline

1 Introduction to ATC

2 The Machine Learning approach to ATC

3 Effectiveness and Efficiency

4 What next?

What next?

Active areas of scientific research currently include

- Semi-supervised learning
  - Allowing classifiers to be trained via a mixture of classified and unclassified documents
- Very large scale (hierarchical) classification
  - Keeping training and classification efficient in the face of very large classification schemes / training sets / sets of documents to classify
- Quantification
  - Optimizing classifiers for class prevalence estimation

Thanks for your attention!

Bibliographic references

S. Chakrabarti, B. E. Dom, and P. Indyk.

Enhanced hypertext categorization using hyperlinks.

In *Proceedings of the 24th ACM International Conference on Management of Data (SIGMOD 1998)*, pages 307–318, Seattle, US, 1998.

N. Fuhr and G. Knorz.

Retrieval test evaluation of a rule-based automated indexing (AIR/PHYS).

In *Proceedings of the 7th ACM International Conference on Research and Development in Information Retrieval (SIGIR 1984)*, pages 391–408, Cambridge, UK, 1984.

P. J. Hayes, L. E. Knecht, and M. J. Cellio.

A news story categorization system.

In *Proceedings of the 2nd Conference on Applied Natural Language Processing (ANLP 1988)*, pages 9–17, Austin, US, 1988.

T. Joachims.

Text categorization with support vector machines: Learning with many relevant features.

In *Proceedings of the 10th European Conference on Machine Learning (ECML 1998)*, pages 137–142, Chemnitz, DE, 1998.

D. D. Lewis.

An evaluation of phrasal and clustered representations on a text categorization task.

In *Proceedings of the 15th ACM International Conference on Research and Development in Information Retrieval (SIGIR 1992)*, pages 37–50, Copenhagen, DK, 1992.

M. Maron.

Automatic indexing: an experimental inquiry.

Journal of the Association for Computing Machinery, 8(3):404–417, 1961.

B. Masand, G. Linoff, and D. Waltz.

Classifying news stories using memory-based reasoning.
In *Proceedings of the 15th ACM International Conference on Research and Development in Information Retrieval (SIGIR 1992)*, pages 59–65, Copenhagen, DK, 1992.

R. E. Schapire, Y. Singer, and A. Singhal.
Boosting and Rocchio applied to text filtering.
In *Proceedings of the 21st ACM International Conference on Research and Development in Information Retrieval (SIGIR 1998)*, pages 215–223, Melbourne, AU, 1998.

F. Sebastiani.
Machine learning in automated text categorization.
ACM Computing Surveys, 34(1):1–47, 2002.

F. Sebastiani.
Classification of text, automatic.
In K. Brown, editor, *The Encyclopedia of Language and Linguistics*, volume 2, pages 457–463. Elsevier Science Publishers, Amsterdam, NL, second edition, 2006.

P. Turney.
Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews.
In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL 2002)*, pages 417–424, Philadelphia, US, 2002.

E. D. Wiener, J. O. Pedersen, and A. S. Weigend.
A neural network approach to topic spotting.
In *Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval (SDAIR 1995)*, pages 317–332, Las Vegas, US, 1995.

Y. Yang and C. G. Chute.
A linear least squares fit mapping method for information retrieval from natural language texts.
In *Proceedings of the 14th International Conference on Computational Linguistics (COLING 1992)*, pages 447–453, Nantes, FR, 1992.