Verification of a human in the loop
or
Identification via the Turing Test*

Moni Naor †

September 13th, 1996

Abstract

We propose using a "Turing Test" in order to verify that a human is the one making a query to a service over the web. Thus, before a request is processed the user should answer as a challenge an instance of a problem chosen so that it is easy for humans to solve but the best known programs fail on a non-negligible fraction of the instances. We discuss several scenarios where such tests are desired and several potential sources for problems instances. We also dicuss the application of this idea for combatting junk mail.

*A preliminary draft.

†Dept. of Applied Mathematics and Computer Science, Weizmann Institute of Science, Rehovot 76100, Israel. E-mail: naor@wisdom.weizmann.ac.il.

1 Introduction

It is quite common now for various companies to offer services on the Web free of charge (or for a promotional fee). These include various search engines (Alta Vista, Infoseek, Inktomi, Lycos, Yahoo, etc.) and shopping arcades and catalogs. An emerging phenomena is that of "meta-services": a program that provides the user with an interface for requesting information (or perform comparative shopping). Given a request from the user, the program accesses several such services in parallel providing each of them with the request. It then processes the information obtained from these services and presents it to the user. Examples for such meta-services are MetaCrawler [7] and Savvy Search [9] for search engines and BargainFinder Agent [1] for comparative shopping. While such meta-services have many advantages for the user, from the service provider point of view they are not necessarily desirable. The reason such services are free is in order to build customer loyalty and for advertisements.Using their output for further processing defeats these purposes, since they lose the direct contact with the user. Therefore it may be desirable for such companies to have a method for forcing a human-being to fill in the forms for the request and not a program[1]. The goal of this paper is to present a scheme that may discourage (unauthorized) meta-services from filtering the user interface of the services they employ. There can be various other settings where one wants to let human users access the resource, but to exclude software robots, or give them a lower priority. One of them, combatting junk mail, is discussed below.

One of the key ideas of Cryptography is applying the fact that there are intractable problems, i.e. problems that cannot be solved effectively by any feasible machine, in order to construct secure protocols. Our proposal is to adapt the way identification is handled in cryptographic settings to deal with this situation. There, when one party $\mathcal{A}$ wants to prove its identity to another party $\mathcal{B}$, the process is a proof that the $\mathcal{A}$ can effectively compute a (keyed) function that a different user (not having the key) cannot compute. The identification process consists of a *challenge* selected by $\mathcal{B}$ and the response computed by $\mathcal{A}$ [2].

What should replace the keyed cryptographic function in the current setting are those tasks where humans excel in performing, but machines have a hard-time competing with the performance of a three years old child. By performing such a task successfully the user's proves that it is human. We envision the following scheme for applying this idea: when a service sends a form to be filled in with the user's request it will also send a "human-in-the-loop-challenge" which will be one or several questions that can be answered easily by any person. When the user fills in his request he should also answer the questions provided as the challenge. Before the service processes the request it should verify the correctness of the answers. The service will not process a query whose attached questions were not answered properly (or will give it a lower priority). The questions for the challenge should be chosen from a large collection of possible questions and will be specific to the user's request, i.e.

[1]An alternative is to ban certain addresses from using the service. The reason this is not sufficient is that in the near future the meta-service may run on the client's machine

[2]This description includes both the symmetric case where $\mathcal{A}$ and $\mathcal{B}$ share the common key that defines a function (the classical Identification-Friend-or-Foe) and the public-key setting where $\mathcal{A}$'s function is defined by a public key so that anyone can verify the correctness, but no one but $\mathcal{A}$ can compute successfully with high probability.

there should be no point in gathering those questions.

We therefore want for our "Turing Tests" a collection of problems with the following properties

1. It is easy to generate many instances of the problem, together with their unambiguous solution. The method for generating the problems can be either internal, i.e. there is a generator that gets as input some random bits and outputs an instance of the problem, or based on external input, e.g. a video camera positioned in a crowded street. It is best if the method for generation does not require human intervention at all. However, if human assistance is needed for creating a model from which several instances are derived, then it is still reasonable.

2. Humans can solve a given instance effortlessly with very few errors. Providing the answer should also be easy, e.g. typing a small number of characters.

3. The best known programs for solving such problems fail on a non-negligible fraction of the problems, even if the method of generating the instances is known. The number of instances in a challenge will depend on this fraction.

4. An instance specification is succinct both in the amount of communication needed to describe it and in the area it takes to present it to the user.

2 Sources for the Turing Tests

We now list a few areas that are a possible source for such problems. They are drawn form Vision and natural language processing.

- Gender recognition - given a picture of a face determine whether it is a male or a female. Since there are only two possibilities the challenge should consists of, say, four pictures and the users should get all of them right. Getting many different pictures for the collection from which the challenge is drawn does not seem difficult, but one should make sure that the are indeed easy for a human being.

- Facial expression understanding - given a face decide whether it is happy or sad.

- Find body parts - Benny Pinkas suggested that the challenge be a picture of, say, an animal and the user should click on its eye. The advantage over all other proposals here is that the number of possible answers is much larger. There should be of course some tolerance for the distance from the correct location.

- Deciding nudity - given several picture determine which one contains the undressed person. Here there is work done in [4] that reports much progress in this area.

- Naive drawing understanding - given a drawing of, say, a house determine what it is from a list of five distinct possibilities. Dan Roth suggested adding "context", i.e. background, to the drawing - this will make it easier for people and harder for machines. Also break the lines.

- Handwriting understanding - given a handwritten word the user should type it. Again, it makes sense to add the kind of noise that people do not have a problem to ignore
- Speech recognition - the challenge is a recording of several words and the user should write them. Given progress in this area, selecting from several possibilities may be too easy; having the user write the result may be too demanding, since there are spelling errors etc.
- Filling in words - Given a sentence where the subject has been deleted and a list of words, select one for the subject. This can be generated more or less automatically from a large corpus. It has the advantage that it is more succinct to represent. However, yet again the progress in solving such problems automatically may be too advanced and using statistical methods is sufficient. Another possibility is to take a sentence and permute the order of the words. The challenge is to determine which of several possibilities is the original one.
- Disambiguation - another problem from NLP (suggested by Dan Roth). The challenge is to figure out to what does "it" refer in a sentence like "The dog killed the cat. It was taken to the morgue." The problem here is that it seems difficult to generate many different examples. Also it may be too demanding on the human user.

3 Remarks

Suppose that the meta-service develops a method that answers correctly 10% of the challenges. Then it would be tempted to try about 10 times in parallel, until its gets one of the challenges it can resolve. In order to prevent this, the set of instances should be chosen as a function of the query, i.e. will be given to the user only after he fills in the form. In any case, it is a good strategy for the service provider not to answer frequent queries from the same source and to follow whether the same query is given frequently in a given time slot.

The scheme suggested here works rather well with advertisements. The file containing the relevant puzzles may include also the advertisements, thus making filtering them difficult. Also the questions in the challenge may be somehow related to the content of the advertisement[3].

Etzioni [3] calls search services like Alta Vista information *herbivores* and the meta-services information *carnivores*. Selberg and Etzioni [7] predict a state where the meta-services work together with the information gatherers by carrying out their advertisements or by sharing the profits. However, it is not clear what motivation the meta-services will have for this Isaiah like ideal [5], unless the herbivores will have the means of protecting themselves. The "Turing Test" approach proposed in this paper provides them with such means.

Another possible advantage of the scheme proposed is encouraging research in those areas that are chosen as challenges. One has to look at the problem of factoring numbers and see the tremendous algorithmic progress made there since it was suggested as a basis for cryptographic protocols [10] to realize the potential.

[3]Interestingly, a Berkeley company called Cybergold plans to offer "rewards" to people who follow links with advertisements.

Regarding some social issues concerning this proposal, it is possible to use in order to create culturally exclusive zones. This may have some advantages, e.g. for keeping children away, but in all likelihood is not very desirable. Therefore, care has to be taken when composing the tests so as not to make them culturally sensitive.

4 Comparison with "Pricing via Processing"

Dwork and Naor [2] proposed a method for combatting junk e-mail and in general for sharing resources when charging for them is either not economical or would not act as a proper deterrent. They suggested that in order for one user to send a message to another user the sender should compute a moderately hard function (taking, say, 20 seconds CPU time on a standard processor) of the content of the letter and the name of the addressee, this way demonstrating that the receiver's attention is important for the sender. The current proposal is also applicable for the junk mail scenario: to send a letter to a user, the sender sends the message and receives a challenge of the type described in the preceding sections that he should answer. The message is forwarded to the receiver's attention only if the sender answers the challenge correctly. In this scenario it is important that the generation of the instances be free of human intervention.

The disadvantage of the Turing Test approach over the one described in [2] is that the protocol becomes a three round one (instead of a single round). Also the proposal of [2] provides for a "shortcut", a way for computing the pricing function efficiently by a trusted agent, say for a reasonable price. This is useful for legitimate mass mailing, say invitations to a party (the deterrence in this case is the amount charged by the trusted agent).

The advantage over "pricing via processing" is that similar commodities are involved - to get the addressee's attention the sender invests some of his own (human) time, and not his CPU time.

5 Further Research

The most intriguing direction of research this paper proposes is whether there are automated Turing Tests: can a computer be the interrogator, i.e. the player trying to establish whether the entity on the other end is a machine or a human. This should be the case even if the program being tested has access to the program of the interrogator (but not to private random bits used to generate problems).

Acknowledgements

we thank Cynthia Dwork, Benny Pinkas, Omer Reingold, Dan Roth and Shimon Ullman for useful remarks.

References

[1] BargainFinder Agent (Andersen Consulting), available http://bf.cstar.ac.com/bf.

[2] C. Dwork and M. Naor, *Pricing via Processing or Combatting Junk Mail*, Advances in Cryptology – Proceedings of Crypto '92, Lecture Notes in Computer Science 740, Springer Verlag, 1993, pp. 139–147.

[3] O. Etzioni, *Moving up the information food chain: deploying Softbots on the World Wide Web*, Proc. AAAI-96.

[4] M.M. Fleck, D.A. Forsyth, C. Bregler Finding Naked People, Proc. 4th European Conf on Computer Vision, 1996.

[5] Isaiah, Chapter 11.

[6] Karov, Y., and S. Edelman, Similarity-based word sense disambiguation, Weizmann Institute CS-TR 96-06, 1996.

[7] E. Selberg and O. Etzioni, *Multi-Service Search and Comparison using the MetaCrawler*. Proci. of the 4th International World Wide Web Conference. Search available in http://www.metacrawler.com.

[8] Y. Moses, D. Reynard, and A. Blake, 1995. Determining facial expressions in Real Time. Proceeding of International Conference on Computer Vision. p. 296-301.

[9] Savvy Search, search available in: http://guaraldi.cs.colostate.edu:2000/form?beta

[10] A. Odlyzko, The future of integer factorization, CryptoBytes (The technical newsletter of RSA Laboratories), 1:2, pp. pp. 5-12, 1995.

[11] A. M. Turing, *Computing machinary and Intelligence*, Mind 59, 433–460, 1950.