What sort of a thing is an AI experiment?

Karen Sparck Jones
Computer Laboratory, University of Cambridge
August 1986

This paper in its final form appeared in *The Foundations of Artificial Intelligence*, (Ed. D. Partridge and Y. Wilks), Cambridge: Cambridge University Press, 1990, 267-281.

Prolegomenon

My concern is with what an AI experiment is, and hence with what AI is. I shall talk about what experiments are actually like, but suggest that this is what they must be like.

Thus is it reasonable to suppose that AI experiments are, or could be, like the experiments of classical physics? I do not believe it is. This is not because we cannot expect the result of a single critical experiment to validate a theory, as we cannot expect a single translation to validate a translation program, for example: we can presumably extend the classical model to cover the case where validation depends on a set of results, for different data. Nor is it because we have not in practice got anything like an adequate predictive theory. I believe that we cannot in principle have the sort of predictive theory associated with physics, because we are not modelling nature in the classical physics sense. I shall elaborate on what I think we are doing, but claim now we reach the same conclusion if we consider the suggestion that we are not in the classical physics position, but rather in that of investigative biologists, doing experiments to find out what nature is like (notionally without any theory at all, though perhaps in fact influenced by some half-baked theory). This is because there is nothing natural to discover. What AI is doing is engineering. While we may indeed have ideas about how to build something so it will work, so we have a predictive theory in a sense, this is not the sort of predictive theory, modelling nature, that physics has. Predicting that people will like what they get, say from a translation program, is not making any specific predictions about the way the translation program models 'real' translation. In other words, AI experiments are engineering experiments serving the designs of task systems, i.e. of artefacts. These systems are artefacts as human task systems are also artefacts. In either case, therefore, we evaluate by performance, so we have no interest in whether the human and computer systems in themselves are the same.

In looking at what AI experiments are like, I shall take the natural language area as my main example. But I shall start from experiments in an area apparently outside AI, because of the light this may throw on AI experiments without supposing one is engaged in something special.

The information retrieval case

I have been led to consider the question of what an experiment is in AI from a very mundane starting point. I have been concerned with trying to build information retrieval (IR) systems, in the sense of document retrieval systems; i.e. with how to index documents and requests and to manipulate index descriptions in searching so as to retrieve documents relevant to the user's need. Indexing and searching are not usually thought of as part of natural

language processing, and hence AI, partly through sheer snobbery, and partly because of the techniques involved. The point here is not that indexing does not require significant natural language understanding, which is false, but that the difficulty of building indexing programs depending on any material understanding of the objects being indexed, i.e. on understanding enough of the contents of scientific papers to pick out and appropriately express the key concepts they embody, has led to the use of statistical information as a surrogate, e.g. to select words with particular distributional properties as index terms. Statistical information about word distributions has a genuine role to play in large scale text handling, as a contribution to text understanding, but it is not adequate as a surrogate.

However there is some automatic indexing work aimed at natural language understanding, or natural language processing, in a more proper sense, and, more importantly for present purposes, some results, if only modest ones, sufficient to justify the belief that one could have natural language programs identifying significant concepts in text. For example we have done work in Cambridge using proper syntactic and semantic analysis to obtain representations of requests from which complex term sources can be extracted and sets of equivalent linguistic expressions generated for searching files of texts. The important question here is then: what is one doing in seeking to build IR systems applying this or some other natural language processing technology?

Consider what is involved in an indexing and retrieval system. We have the givens, i.e. the data variables. Even without taking into account the properties of users as these determine relevance judgements, and confining ourselves to the more accessible parts of the data, namely documents and requests, as information objects, we have a large number of variables, some with many possible values. For example, for document texts we have more obvious variables like language, subject, length, specialisation, type, etc, and similarly for requests. Particularly from the document point of view, these are properties of objects both as individuals and as members of (possibly very large) collections. Some of these properties, like subject, can clearly have many values. We have further to take types of need, e.g. for a few relevant documents or all of them, into account. There are also less obvious properties like the well-formedness of requests. Similarly, for indexing and searching, we have many system parameters, with many possible settings, for instance the indexing language, form, length etc of index descriptions, searching strategy, matching function, and so forth. These too may have many settings, for example the matching function as the number of shared terms or some other scoring coefficient.

As this list shows, an IR system is not a natural entity: it is an artefact. But it is no more, or no more fundamentally, an artefact than many other language-based information systems, whether these are the private systems of individuals or community systems. The important point is that an IR system is a system which is designed to solve a problem, that of describing the content of documents and the nature of users' requests so that a match between descriptions is correctly deemed to indicate a document is relevant to the user's request (or more properly, to his underlying need). Important additional constraints seen most clearly in designing IR systems for community use are that one is unavoidably designing for average use, and coping with the impact of scale on strategy performance, given the normal requirement to select the very few relevant documents from the many non- relevant.

The problem, then, in designing IR systems is fighting one's way to fitting parameter settings to variable values, when the variables themselves may be substitutes for inaccessible system components, e.g. words for ideas, request texts for user needs, and when the variables also interact, as do the parameters, for example term specificity and description exhaustivity.

There are further daunting problems, in evaluating system designs, of finding appropriate measures of performance, especially average performance, given the differences between requests, and in testing of sampling (one cannot judge the relevance of every document for every request), and of significance testing. Here we are dealing with the second-order parameters of experiments. (For a fuller discussion of these questions see Sparck Jones 1981.)

None of this is indeed special to IR system design as opposed to other engineering system design. The material point here is that one is dealing with processing information expressed in natural language, supposedly an AI activity, and so with the implications for building AI systems of such engineering design enterprises.

These are, first, that the only way to try to figure out whether one's system is doing what one intended, i.e. that the observed performance is attributable to the interaction between the assigned parameter settings and perceived variable values, is by rigorous, controlled experiment. The complexity of IR systems, and the limitations of our understanding of how they behave, together mean that whatever theory we have of how to build them needs intensive and extensive testing. The number of variables and parameters, and in some cases of their possible values, makes IR experiment a grinding business of systematic variation with not much confidence in the certainty and uniqueness of the value- setting correlations and hence performance-factor attribution. The second, more significant, point is that strictly there are no right answers for the system to deliver: the set of relevant documents for a query is not well-defined. In practice something may be said about it, for example that whatever else, this document is relevant to this query, which is sufficient to drive evaluation in an empirical spirit; but relevance is a furry concept, which limits the definition of experiments. One feature of relevance, for example, is that the user's view of what is relevant can change as he proceeds through the system's output. Thus the evaluation problem for the IR system designer is ultimately one of principle: he may be able to approximate the set of relevant documents well enough for useful testing, but he cannot specify his goal sufficently precisely for it to give him an absolute check on performance.

How does this reality of experiment in IR system design apply to AI in general? First, as suggested earlier, it is not obvious that IR itself is not an AI activity: determining what texts are about, and indicating their essential meaning, implies language understanding; similarly, relating one text to another, whether that of one document to another, or of a document to a request, i.e. searching for information, implies language based reasoning. The determination, representation and use of linguistically expressed knowledge is a characteristic activity of AI. This is not to suggest that current automatic indexing and searching practice comes anywhere near this, though some research is closer. It is therefore useful to consider here another linguistic task for which current research claims natural language understanding techniques, and also a non-linguistic AI task. Do either of these presuppose experiments of the kind described?

The summarising case

Consider summarising as a natural language processing task. Summarising is of interest here in that though I have described indexing in a manner which means it merges into summarising, one is in principle often looking for something altogether richer than an index description (compare the abstract of a scientific paper with an index description, even one in the form of a phrasal subject heading, let alone a simple list of terms), and certainly aiming at something richer than the product of current automatic indexing practice. Producing a worthwhile abstract clearly requires natural language understanding of a serious sort, i.e. an

extremely complex program.

This program will have very many elements, for example syntactic category set, semantic feature system, parsing strategy, focus determination mechanism, anaphor resolution procedure, recovery methods for ill-formed input, etc, applicable to text interpretation, with analogous processes for generation, as well as a central summariser; and these elements may take many forms, for example there are many possible semantic feature systems. In such complex systems we have to allow for considerable elaboration of the notion of parameter and setting, but it applies nonetheless. There are also many data variables, with their possible values, to take into account, for example language, text type, sentence type etc, again implying a much more extensive structure of data characterisation than the IR case, but not one different in principle. Similarly, the function summaries should serve is the system goal analogous to the IR system requirement to deliver relevant documents. A summarising system is a task system, like an IR system, though what purpose summaries, as opposed to index descriptions, serve is the crux, as the points which follow show.

The many factors involved in a summarising system mean that, as in the IR case, it may be far from obvious how features of the system's outputs are to be attributed to particular combinations of given variable values and chosen parameter settings (whether the specification of values and settings is supplied or system selected is irrelevant). It is easy to identify manifestations of bugs in natural language processing programs, for instance incorrect inflections or wrongly resolved pronouns, but as the second example suggests, it may not be so obvious with a complex anaphor resolution procedure involving, say, linguistic focus mechanisms and non-linguistic inference on sentence representations delivered by syntactic and semantic processors, where the source of the trouble is. This is particularly likely in summarising, where the results of intermediate processing, for example focus determination or the extraction of presuppositions from input sentences, are not necessarily carried forward to the output but where failures in them may influence the output. Thus a 'wonky' pronoun in the output summary text could be attributable to a variety of causes including faulty input sentence interpretation, defective summarising, or an inadequate generator. But this kind of opacity is not special to AI programs, and though practically tiresome is not theoretically interesting.

The more important point is that a summarising program can apparently fail in a way which is not attributable, crudely, to a bug, i.e. we are talking about performance in a looser sense: a summary can be a good one, or a fair one, or a poor one, and we may find one better than another. It is true we may claim in some instance, for example if we can point to a missing concept, that a summariser has failed in a fairly straightforward, somewhat buggy sense. But we can talk about a poor summary or say that one summary is less good than another without being able to point to anything comparatively definite like a missing concept. More seriously, we could get two alternative summaries, from different program designs, with, for example, different relative emphases on different concepts, without it being at all obvious which one is better, as indeed one program by itself could deliver a perfectly satisfactory looking summary. What then is right or correct? We can seek to apply the kind of evaluation criteria used for the evaluation of human abstracting, like "Are the main points covered?". But these criteria are typically rather high- level or crude ones which are difficult, if not impossible, to correlate with specific components of the system and their parameter settings, for instance the style of lexical entries or even, if individual lexical entries are treated as parameters, the content of a specific entry. Thus as the criteria are very general, they can only be used in a very undiscriminating way.

This is unfortunate but inevitable: these vague criteria reflect the fact that there is no such thing as an objectively right or correct summary of a text. We can talk about an acceptable (because useful) summary, but this is a very weak form of program control. With summarising, therefore, we are up against the fundamental problem that the required form of processing can only be put in vague terms like "pick out the essentials", with the consequence that we cannot get the precise measure of output quality needed to feed back to program design. The same problem of what makes one summary better than another also appears, for example, in comparing a more summary summary with a less summary one, or even so-called types of abstract, indicative, informative and evaluative, with one another. It may be as unobvious that one length or type is better than another as it is that one alternative of the same length or type is better than another. Thus even if one supposes that an evaluative abstract is superior to an indicative one, say, this belief needs justification in relation to some manifest summarising purpose.

Candidate formal criteria for abstracting, like requiring proper entailment relations between source and abstract text, do not, quite apart from the formal problems this proposal itself involves, do anything to solve the key problem either, which is determining the important items we want to constitute the abstract. Attempts to select these by looking, for instance, at the entailment productivity of propositions, in turn get into all the difficulties associated with a reduction to counting, where any particular formula seems an arbitrary way of capturing a general concept. The root problem of summarising is just that summarising involves selection, or alternatively elimination, so it is difficult compare source text and abstract, especially when many possible different selections can be made. The only base for evaluation is thus the functional one.

The significant point about a summarising system, therefore, is not so much that its greater complexity makes it much harder to attribute output phenomena to their program sources, though this is true, but that there are problems about determining the quality of the output. There is a significant difference between IR systems and summarising systems as I have described them, residing in the way summaries are used. This may only be a matter of degree, but it is a large and so important one. Though I described the core process in the document retrieval case, namely indexing, so as to emphasise its similarity with abstracting, index descriptions in retrieval services are designed for machine searching, which is not true of summaries, certainly in practice, and even in principle. The scale of retrieval services like DIALOG implies machine searching, and is indeed their justification. The point about a summariser is that its output is for the human reader, in just the same way as any other text is primarily for its human reader; i.e. searching delivers relevant documents to the user: reading delivers nothing so definite. The problem then is what comes of performance measures. We have an independent means of evaluating the quality of indexing in a retrieval system because we can measure the system's ability to deliver all and only relevant documents, even though, as pointed out earlier, our means are necessarily limited. A retrieval system's claim that the documents it deems relevant to a request, because its description matching function is satisfied, are in fact relevant can be tested, if only crudely.

When we look at an abstract on the other hand, and say it is a good one, on some abstracting criteria, as we may also look at index descriptions, this assessment cannot be anything like as exigent as the retrieval test. Of course whether retrieval searching is done mechanically or not is not the essential point, it merely emphasises it. Thus even if index descriptions are read rather than mechanically searched, this is still to achieve a relatively defined purpose, namely the identification of relevant documents. The problem with summarising being for the human

reader is that this is a manifestation of the fact that there is no well-defined task associated with the reading for measuring the effectiveness of the summary; we have no clear functional basis for evaluation. Exploiting summaries in further mechanical processing, whether in the form of an 'internal' representation of their content in explicit natural-language text form, does not affect the issue. The summaries are then being used for some purpose which provides a context to evaluate them. The evaluation may be more or less rigorous, according to the system's function, but only if the purpose is unusually narrow will it be possible to have clear eavluation of something as intrinsically complex as a summary.

Thus my argument is that the complexity of natural language objects (or their representations) is generally correlated with the indefiniteness of their uses. The more indefinite the need the more complex it actually is, and hence the richer and less easy it is to define the object required to satisfy this need. Indeed the problems in the IR case with the notion of relevance are a reflection of the fact that even here the user's need, and hence the system's task, are not very well defined. The point about summarising is thus primarily that it much more clearly poses the problem of system evaluation without a well-defined task, and therefore a means of establishing whether the system has delivered the right or correct answer. The only form of evaluation that seems possible in cases like this is that the system delivers acceptable output: and this is a fairly weak form of evaluation.

Non-linguistic cases

Similar problems, though with differences of emphasis, apply to other language processing system tasks like translation. However a more serious question is whether the same issues arise in other non-linguistic areas of AI, i.e. whether there are other situations where we have a complex artefact and necessarily no very precise base for performance evaluation, so one has to proceed essentially on a trial and error basis weakly supported by intuitive judgement or socio-economic measures of effectiveness. It is not difficult to see at least some robot applications and even more some expert systems as of this kind. How are we to say whether a robot vehicle wandering round the surface of a planet is performing optimally: what would optimal performance be? We may assert a robot fails if it falls down a hole, but how can we say it has picked up the best set of rock samples? The same goes for an expert system like a holiday advisor (I am concentrating here on the non-linguistic decision- making apparatus of the system). With a medical expert system we may have something like an independent performance measure: its decisions are those of a panel of doctors; or the patient lives. 1But for a holiday advisor it is not obvious that there is anything like a specific independent measure: even if the advisor performed the same as a clutch of travel agents, how relevant is this to the quality of the advice; and if it is not, how are we to define what satisfactory advice is?

Conclusion

The conclusion is therefore that the role of experiments in AI is to try out designs for engineering artefacts, to see how well some system will meet some need. The detailed methods adopted can vary: one can use customer samples, or simulation, for example. This implies a performance measure related to the system's purpose, which may be more or less easy to find. But it is a measure of acceptability not of truth. The fact that one is dealing with artefacts, moreover, does not imply that these systems are distinctively different from the human ones being emulated: they too are personal or social artefacts. Thus in the language case, as one

characteristic example, we should not overrate the objective reality of human processing as something to measure a program's internal operations against; we can only evaluate systems by their performance, and that very loosely for activities like summarising. But evaluating by acceptability is perfectly respectable. What humans do is in the real sense ad hoc: they build systems to work well enough. There is therefore no reason to require program builders to do anything different.

Sparck Jones, K. (ed) *Information retrieval experiment*, London: Butterworths, 1981.