

MIMes and MeRMAids¹: On the Possibility of Computer-aided Interpretation

Geoffrey Rockwell
McMaster University
georock@mcmaster.ca

Abstract

One problem in humanities computing is the automatic interpretation of electronic texts. Can computers interpret texts or do they aide in the interpretation? In this paper we propose variants on the Turing Test to imagine if computer-assisted interpretation is possible. We argue that a way forward for developing models for successful machine interpreters is to work with regular expression recognition and manipulation, which is built into most programming languages in a common fashion.

KEYWORDS: Humanities Computing, Computer-Assisted Text Analysis, Interpretation.

1. Language and Interpretation

Can machines generate interpretations of texts?

Willard McCarty in a post to the discussion list HUMANIST asked what the great questions in humanities computing might be.² If one takes his question as the first, which it is, what follows is a proposal for the second question and that question is whether it is possible to design machines that can automatically interpret texts. This essay sets out to formally describe the problem and suggest games for answering it which, of course, will never answer the question, but will help us learn more about interpretation.

For our purposes I am going to define a language the way it is defined in computing as the set of legal strings that can be generated from a character set. Given a character set Σ and rules for generating legal strings from that character set one can build an engine that would recognize whether some string is a legal member of the language.

Such a formal language is not necessarily a human language, but given an alphabet used in a human language it should at least include all the strings that would be part of that human language. The rules for the generation of legal strings are the grammar of the language. From the

grammar one can develop an algorithm for recognizing strings that are members of that language and from that algorithm one can build a machine (in theory) that could recognize a legal string.

There is another type of algorithm that can, given a legal string (a legal expression in the language) generate another legal string in the language. I will call the machines that implement this set of algorithms Text Transformation Engines (TeTE). A TeTE takes a string in a language as input and outputs another string in the language based on rules for transformation.

Original String -> TeTE -> Transformed String

A subset of the set of possible TeTEs is that set of engines which can output a string that would be recognized as an “interpretation” of the original string. I am going to call such an engine a MIME for Meta-Interpretation Machine. I call those engines that can interpret any text “Meta” machines. (We can also imagine machines designed to interpret only particular texts and those I would call a LIME for Limited Interpretation Machine.)

Thus the problem of interpretative machines is whether there are any MIMes. A pragmatic approach would be to look around and ask if there are any machines that have produced something we would agree is an interpretation of a text (LIMes) and then ask if these machines can interpret any reasonable text (MIMes). The problem with the pragmatic approach is that a true MIME should be able to reliably produce an interpretation given any legal input string; anecdotal evidence that a machine has produced some interesting interpretations of a string does not mean that it can reliably produce interesting interpretations of all texts for all time. How would we prove that a candidate MIME could handle any text including those not yet written like the very interpretations that MIME will produce when tested?

2. Galatea Tests

Alan Turing

More importantly we have the difficulty of agreeing whether some output constitutes an interpretation let alone an interesting one. If interpretation is defined as a human activity no machine, by definition, could be a MIME. The most we can hope for is a TeTE. This is the problem Turing faced when asking about artificial intelligence. Like Turing I am going to side-

step the question of what an interpretation is by proposing a game, which I will call the Galatea Test after the novel Galatea 2.2 by Richard Powers where a novelist and computer scientist develop an AI that can answer MA level questions about literary texts. Here is the Galatea Test:

- a. A judge is chosen who self-identifies as being able to recognize interpretations. This judge is connected to a human interpreter and a machine interpreter, either by a terminal or by some other system that can guarantee that the judge knows nothing about his/her interlocutor.
- b. The judge starts a turn by submitting a legal string (defined as belonging to the language L generated by the character set Sigma according to the grammar G) to both interpreters. We can, to simplify things define a legal string as a literary or poetic work which has been interpreted before.
- c. Both interpreters prepare an interpretation which is returned simultaneously to the judge, when they are both ready.
- d. The judge decides which one he/she believes to be by the human interpreter (and consequently which by the artificial interpreter or MIME candidate.)
- e. If, over a number of these turns (steps b-d), the judge guesses right 50% or less of the time then we can say that the machine interpreter has passed the Galatea Test for that judge and those texts used by the judge and therefore we can say that the artificial interpreter is a provisionally a MIME.

Those familiar with the Turing Test and its critics will point out a number of problems with such tests.

- i. The test doesn't tell us whether the MIME interprets the way a human would whether or not the outcome is indistinguishable. In fact, it is likely that a successful MIME, as Deep Blue showed for chess, would probably pass by using non-human techniques that involve "brute" force computation or a vast database of amusing interpretative moves. At best we have a simulation of interpretation (the

act) not the act itself. See Searle's Chinese Room Paradox, problems about consciousness and so on.

My response to this is that we are not interested here in the act of interpretation but the resulting interpretation and whether machines can produce interpretations. The Galatea Test is designed to test whether a machine can be designed that will produce interpretations (output) indistinguishable from those humans produce.

ii. Another criticism of the test is that it misses the point. As Alan Kay points out, we don't want to automate what we like to do, we want to automate what we can't do or don't like to do. Interpretation is one of those things we generally like to do (except when under a deadline as an undergraduate), therefore the test doesn't test for what we should be interested in which would be whether MIMes can be designed that either help with the boring parts of interpretation or which generate interpretations that humans would not generate. This raises the question of whether we would recognize an interpretation that was not human as an interpretation rather than machine noise. The paradox that corresponds to the question of machine interpretation is that we might not recognize an artificial interpretation even if we got one from a MIMe. In other words, true MIMes may not mimics.

My response is that trying to simulate the richness of human interpretation is part of understanding what machines can do and what we don't want to bother having them do. The Galatea Test is designed to answer the original question – are MIMes possible. That question I believe is interesting to humans even if the resulting machines may not be. Further, it is likely that, should we succeed a producing a MIMe, humans would devise new ways to interpret that could not be simulated, at least by the successful implementation. Imagine what would happen to human interpretation if we were successful? Oh no... time to fire all the literary critics.

iii. A third problem is due to the limitations of the test. We can imagine languages and therefore interpretations that involve gestures, graphic designs and so on that would not be possible to test. The Galatea Test, like the Turing Test privileges linguistic text and the interpretation of text, and in particular text in formal languages as defined so innocently at the beginning. That said, this is a reasonable limitation for the moment since what we want to determine is whether there is a MIMe at all over a language as defined. I think it is possible to redesign the test to include more inclusive definitions of language and text as long as one can formally specify the language and isolate the judge so that they have no mechanism for judging the artificiality of the interpreters other than the interpretations generated in that language. There is no a-priori reason why robotic MIMes capable of handling and generating gestures as part of an interpretation would be any more difficult than a textual MIMe.

iv. The most significant problem with this Test is that we are unlikely to get anything resembling a successful MIMe for years, if at all. While the AI community seems to be willing to put up with the Turing Test and its limited form in the Loebner Prize with a certain sense of amusement or at least resignation, humanists are likely to lose interest in the Galatea Test and therefore consign the problem to the dust heap of trivial problems. Another way to put this is to acknowledge a variant of problem ii. – ie. that the Busa Test sets such a high standard that taking the test seriously is unlikely to ever result in interesting Humanities Computing research whether or not it is a fundamental problem. One way forward would be to offer money, as Loebner did, for a successful MIMe. (In the spirit of the humanities I will instead offer a copy of the complete works of Plato.) Another is to imagine limited versions of the test which could provide a research agenda for the near future.

3. MIMeAides and the Busa Test

Can machines generate aides to interpretation?

If we don't need computers to do what we like to do and if interpretation is something we like to do, then a limited version of the problem would be to ask if there are machines that can aide in interpretation. Such machines would not generate interpretations, but would generate aides to interpretation. An obvious example would be a concordancer which given a text would produce a concordance which a human can use to interpret the original text. Setting aside the question of what the difference is between an aide to interpretation and an interpretation itself, such aides, which we will call MIMeAides, are closer in theory to what we should be satisfied with, but they are more difficult to test for since their ability to aide in interpretation is itself open to interpretation. A limited form of the Galatea Test where a judge had to identify the human generated interpretative aides and the computer generated aides would be difficult to set up because something could be a genuine aide and also be clearly generated by a computer. Rather, a more complex doubled test can be imagined which we will name after Roberto Busa, a pioneer in humanities computing and computer concordancing.

- a. A judge is chosen who self-identifies as being able to recognize interpretations. The judge selects a number of texts for interpretation.
- b. Two interpretation teams are set up. Team A is made up of a human interpreter AH and a human interpretative helper AHH, while team B is made up of a human interpreter BH and a machine helper BAH. Both human interpreters should not be familiar with the text strings the judge chooses for interpretation.
- c. The judge starts a turn by submitting a legal string and a question of interpretation that are passed to the two aides (AHH and BAH.) In effect the question and legal text can be treated as a single text with two parts.
- d. The two aides generate an interpretative aide in the form of the original question followed by a text that aims

to help the interpreters. These aides cannot include more than a portion of the original string in its original form. When the two aides are generated they are simultaneously passed to the two interpreters (AH and BH.) Note that the original text string is not passed to the interpreters.

e. The two interpreters are given a limited amount of time to generate an interpretation which is an answer to the question of interpretation using the received aides. The two interpretations are returned simultaneously to the judge.

f. The judge decides which one he/she believes was by the interpreter helped by the human aide.

e. If, over a number of these turns (steps b-f, during which the human interpreters switch aides), the judge guesses right 50% or less of the time then we can say that the machine aide has passed the Busa Test for that judge, those texts and questions, and those human interpreters, and therefore we can say that the machine aide is a provisional MIMeAide.

A number of the elements of this test, like the number of turns one needs and the amount of the original text that can be excerpted by the aide, remain to be worked out to have a formal test. I propose that such a test be first played in order to determine the optimal rules given the complexity of the doubling. Only then could it be used to test whether there are MIMeAides. As such it should be called the Busa Game not Test. Such dialogical games have, after all, a long tradition in the humanities as what in the social sciences they call method.

The virtue of the game is that we can play it with many of the existing text analysis tools and through playing it, learn about interpretation and the tools. At this point in the discipline, we can actually make progress on MIMeAides. The game, even if treated only as thought experiment, gets at the heart of what matters to computing humanists: can computers help us understand our textual history?

4. Research Agenda: Developing a MeRMAide

How we go about developing candidate MIMeAides? One approach is to build on what we know about regular expression recognition and processing. The final section of this paper is a proposal for how we can collaboratively start building and comparing MIMeAides.

Built into most text processing languages like Perl, Python, and Ruby, is the capacity to do regular expression recognition and manipulation. Using what has become a standard syntax (that interestingly is embedded in very different programming languages as a meta-language) for describing a regular expression one can describe patterns that you want recognized and processed. Regular expressions are ways to describe patterns in items of sets for matching and processing. They can also be used proactively to provide a grammar for a “regular” language. In this context one can think of Regular Expressions as rules for what are legal strings in a language generated from a given character set.

Languages defined by regular expressions are “regular” languages and have been studied by linguists and computer scientists.

For our purposes what is important is that any candidate MIMeAide is typically going to have to do the following:

1. Analyze the Text. Typically a MIMeAide will have to recognize complex patterns in the full text. This step might actually be constituted by a number of steps that break down the text into parts (tokenize) and then recognize parts using rules. Both the breaking down and the pattern matching can be described with regular expressions. Something as simple as breaking a text into a set of lines is just recognizing for the line feed character.

2. Synthesize an Aide. Once the text has been analyzed a candidate MIMeAide should synthesize a new text that is an aide to interpretation. The synthesis may, as in the case of a concordance, involve reassembling selected parts into a new text, but it could also involve more sophisticated generation techniques or just an enormous library of witty things to say whatever the input. Regular expressions as they are implemented in most text processing languages can actually do more than just recognize a pattern, they

can also manipulate it. The regular expression syntax of Grep has been implemented in most modern languages and that allows one to specify not only what to look for, but what to do with it – in other words, how to synthesize something new with what you find.

In short, we can with regular expression languages, describe a class of ways of analyzing a text and synthesizing a new text from the original. If we can call any machine that uses regular expressions to process data a MeREMe (Meta-Regular Expression Machine) then there is subclass of MeREMes that overlaps with MIMeAides called MeRMAides (Meta-Regular expression Machine Aides). MeRMAides, besides being a hybrid or monster that combines the human and the other, would be those machines that use regular expressions to analyze and synthesize texts in order to aide us in interpretation. MeRMAides are only one model for how we might build MIMeAides or MIMes, but it is an accessible model that builds on what is effectively the standard across programming languages for pattern recognition. Regular expressions also have the virtue that they can be compared across implementations, exchanged to be used in different implementations, and wrapped in other code that can handle input and interface. Importantly, regular expression processing is widely supported not only by standard tools and programming languages but a variety of free tools.

5. Conclusion

Mark Olsen, in conversations at the ACH/ALLC, has repeatedly challenged me to demonstrate that humanities computing has contributed to research in other disciplines. Over the years he has moved from arguing that computing can only assist research into text corpora to doubting if even then one can get useful results from a machine. This paper proposes a couple of formal tests to determine whether machines can be designed that can produce interpretations, which I take to be the essential problem in Olsen's larger challenge. Only if we can create machines that can assist in interpreting textual evidence can we then tackle the challenge of designing machines that can produce significant research interpretations.

I am not as disappointed as Mark Olsen is in Humanities Computing because I frankly don't care if we ever generate interesting results for other disciplines -- would they care if we did? Humanities Computing is

its own discipline with its own problems such as the problem proposed here; problems we are just beginning to articulate in opposition to those of other fields. What matters to the discipline is whether we can pose questions that are unique to Humanities Computing - questions that interest us and around which we can do research as a community. That is what “pure” Humanities Computing is about, and incidentally how the discipline might actually be an inspiration to other disciplines rather than a servant. The core research of Humanities Computing should not be designed to be of use to other disciplines; it should focus on what it is to be human when extended by digital machines -- what it is to play with mimes and mermaids.

So, what are the other problems of Humanities Computing?

Notes

¹ MIMes and MeRMAids” first appeared in French as “Des MaMI et des MaMER: Sur la possibilité de l’interprétation assistée par ordinateurs” <<http://www.uottawa.ca/academic/arts/astrolabe/articles/art0040/Interpretation.htm>> in L’Astrolabe a peer reviewed online research site edited by Michel Lemaire. It appeared in 2003 in a translation by Stéphanie Posthumus. The paper is based on a keynote address at the Inaugural Canadian Symposium on Text Analysis held at the Université de Montréal in 2002. A preprint version is also available in PDF form at Geoffrey Rockwell’s web site <<http://www.geoffreyrockwell.com/publications.html>>.

² The call came in a message to HUMANIST <http://lists.village.virginia.edu/lists_archive/Humanist/v15/0010.html> (Accessed 11-22-2002).

Date: Thu, 10 May 2001 06:57:57 +0100

From: Willard McCarty willard.mccarty@kcl.ac.uk

Subject: birthday presents

Dear colleagues:

Many thanks on behalf of everyone for the birthday messages. Only one person I know actually orchestrates birthday presents for herself, but the strategy seems to work, so I thought I’d follow her example here. May I suggest, then, that you send to Humanist a birthday present in the form of a question or statement of a problem concerning humanities computing that bothers you most? Some piece of mental grit that gives you sores every time you go on your mental way. Wonderful gift for Humanist.

Yours,

WM

Works Cited

- Busa, Roberto. (1980). "The Annals of Humanities Computing: The Index Thomisticus." *Computers and the Humanities* 14.2: 83-90.
- McCarty, Willard. "Birthday Presents." Online posting. 10 May 2001. Humanist Discussion Group. 22 Nov. 2002. <http://lists.village.virginia.edu/lists_archive/Humanist/v15/0010.html>.
- Powers, Richard. (1996). *Galatea 2.2*. New York: Harper Perennial.
- Rockwell, Geoffrey. (2003). "Des MaMI et des MaMER: Sur la possibilité de l'interprétation assistée par ordinateurs" *L'Astrolabe* Ed. Michel Lemaire. Trans. Stéphanie Posthumus: <<http://www.uottawa.ca/academic/arts/astrolabe/articles/art0040/Interpretation.htm>>. Preprint: <<http://www.geoffreyrockwell.com/publications.html>>.
- Turing, Alan. (1950). "Computing Machinery and Intelligence." *Mind* 59.236: 433-460. Rpt. (1997). "Computing Machinery and Intelligence." *Mind Design II: Philosophy, Psychology, Artificial Intelligence*. Ed. John Haugeland. Cambridge, Massachusetts: MIT Press, 28-56. 22 Nov. 2002 <<http://www.loebner.net/Prizef/TuringArticle.html>>.