Turing's conceptual engineering

Abstract

Alan Turing's influence on subsequent research in artificial intelligence is undeniable. His proposed test for intelligence remains influential. In this paper, Turing's conception of how to understand intelligence is analyzed as an instance of conceptual engineering that rejects the role of the previous linguistic usage but appeals to intuition pumps instead. Even though many conceive of his proposal as a prime case of operationalism or behaviorism, it is more plausibly viewed as a stepping stone toward a future theoretical construal of intelligence. In addition, his own conceptual network is analyzed through the lens of distributional semantics over the corpus of his written work. As it turns out, Turing's conceptual engineering of the notion of "intelligence" is indeed quite similar to providing a precising definition with the aim of revising the usage of the concept.

0. Introduction

Alan Turing’s influence on the development and progress of computer science and artificial intelligence cannot be overestimated. His accomplishments in these areas remain also striking in regard to general philosophical issues such as the very possibility of artificial minds. Simply, Turing’s proposal defined the field of inquiry. Despite numerous objections, the Turing Test remains the most influential way to consider the possibility of artificial intelligence.

In what follows, the focus is on the foundational 1950 paper that presents the famous test. My goal is to understand Turing’s proposal as an exercise of conceptual engineering aimed at revising our notions. Conceptual engineering is the project of revising, or ameliorating our concepts, rather than simply analyzing them (Blackburn 1999; Cappelen 2018). My analysis is complemented by a more “distant reading” of Turing’s oeuvre: by analyzing general patterns of his language use, parts of his conceptual framework are sketched. To accomplish this aim, a corpus of Turing’s writings was compiled and then analyzed through text mining.

After presenting the assumptions used in this paper, Section 2 focuses on Turing’s argument against relying on common sense in debates over thinking machines. It is argued that Turing proposes an alternative framework: instead of conducting conceptual analysis, Turing engineered the notion of intelligence through a thought experiment. However, the aim of the experiment in his engineering remains somewhat debatable. In particular, the question is whether the Turing test provides an operational definition of intelligence. Subsequently, why the notion of intelligence was preferred by Turing over the more traditionally entrenched notion of mind is discussed. This leads to a general discussion of the semantic fields of crucial psychological terms in Turing’s corpus of his published work.

Text mining allows me to provide tentative answers to two questions: (1) Why did Turing rely on the notion of intelligence, rather than that of the mind? (2) What role does the Turing test play: is it an operational definition or not?
1. From close to distant reading of Turing

The traditional way to engage intellectually with philosophical arguments is to meticulously reconstruct their historical context and conceptually analyze their contents. This approach has become deeply entrenched in philosophy and clearly has merit. In this paper, however, it will be complemented with a more “distant” way of reading. It is not meant to replace close reading; instead, it can provide additional context to understand the overall project in which Turing was engaged.

Over the past few years, huge progress has been made not only in natural language processing (NLP) but also in digital humanities, which, to a large extent, rely on NLP techniques. New developments in deep neural networks led to a breakthrough in the performance of NLP in multiple applications, from machine translation, through speech processing and question answering, to text analysis and text mining. Text mining aims at extracting structured information from unstructured textual data. While there are multiple methods one could use to gain new perspectives on historical work through text analysis and text mining, such as topic modeling or diachronic study of the frequency of certain terms, in this paper, distributional semantics is the approach of choice, accompanied with a mere glimpse into the topical structure of Turing’s work (which is the aim of topic modeling). According to distributional semantics, the meaning of a word is determined by what accompanies it in a large body of text.

This approach to semantics has recently become influential in NLP, in particular thanks to its use in groundbreaking language models such as word2vec (Mikolov et al. 2013) and Glove (Pennington, Socher, and Manning 2014). Its roots stretch back, however, historically much deeper. Already in the 1950s, John Firth advocated for the principle “You shall know a word by a company it keeps!” (Firth and Palmer 1968, 179). His idea was that lexicography presents the meaning of words through their collocations. For the purposes of this paper, a collocation is a sequence of words that occur together more often than would be expected by chance alone (for a recent review, see (Gablasova, Brezina, and McEnery 2017)). This is because, he stressed, meaning is indicated by habitual collocations that correspond to common patterns of usage.

While the lexicographic practice of providing citations in dictionaries to illustrate word senses is fairly innocuous, Firth’s approach is more radical because it relies only on intertextual relations. His assumption seems to have been that meaning is reducible to intertextual relations between words. Hence the stress on collocations. These relations are now studied through computational models of language but the idea remains the same. The resultant radical feature of this approach to meaning is that it seems to have no place for reference.

To appreciate the radicality of this founding assumption of distributional semantics, it is sufficient to note that according to Firth’s approach, the symbol grounding problem (SGP), formulated by Stevan Harnad (Harnad 1990), is a pseudoproblem. The SGP is the question: “How can the meanings of the meaningless symbol tokens, manipulated solely on the basis of their (arbitrary) shapes, be grounded in anything but other meaningless symbols?” Firth’s
answer seems to be: “This is not a reasonable question at all.” Interestingly, Harnad relies on the analogy with a dictionary:

```
Suppose you had to learn Chinese as a second language and the only source of information you had was a Chinese/Chinese dictionary. The trip through the dictionary would amount to a merry-go-round, passing endlessly from one meaningless symbol or symbol-string (the definiens) to another (the definiendum), never coming to a halt on what anything meant. (Harnad 1990, 339)
```

A radical supporter of distributional semantics might respond that this is not a merry-go-round, but a way to establish a complex multidimensional vector representation of meaning. In modern word embedding models, meaning is represented in highly multidimensional vectors (the size of which depends on the number of vocabulary words to which all other words are related). By appropriately setting up these vector values, one can establish how close or related words are: the company of a word is what is closer to this word in this multidimensional space.

While demonstrably, this approach has allowed NLP to flourish in recent years, thanks to rapid developments in deep neural networks, the premise that meaning is entirely reducible to symbol token embeddings is not required to explain its success. Moreover, one can still believe that even the largest word embedding models remain much more brittle than human language users when presented with real-world tasks. This can be easily evidenced by the fact that these models cannot pass the Turing test.

Far from presupposing that the symbol’s embedding constitutes the entirety of its meaning, in this paper, I assume that one can (or even, perhaps, should) study collocations to better understand the conceptual structures. In other words, what is assumed here is that collocations shed light on usage; this is the innocuous assumption made every time one looks up a dictionary entry to read definitions along with some typical citations.

These patterns are what sometimes remains implicit in the body of text and can be easily missed. Just as Harnad did not appreciate that the mere study of how shapes go together can be used to build a vector representation to study their semantic relatedness, we might not perceive certain semantic relationships by closely reading Turing’s work. Close reading is indeed indispensable in following his line of argument but insight into more general usage patterns can provide more context. This is what “distant reading” is about: finding the overall semantic structure.

2. Why not common-sense intuition?

In this section, I will first interpret the Turing test as a thought experiment for revising our notion of intelligence. Then, I will contrast this experiment with another instance of conceptual engineering, found in his foundational paper on computation (Turing 1937), to show that its goal is different. Finally, I discuss two questions that remain undecided by the close reading of Turing.

Turing’s immensely influential paper on computational intelligence starts with a discussion of an alternative approach to the conceptual analysis of the question of whether machines
can think. What makes his argument particularly interesting is that it seems to reject both simplistic philosophical conceptual analysis and the distributional approach proposed in the previous section. Thus, let us closely read this opening passage of his paper:

I propose to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, 'Can machines think?' is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words. (Turing 1950, 433)

In this passage, Turing’s argument seems to be as follows:

1. Answering conceptual questions involves the study of lexical definitions of the meaning of component terms.
2. Lexical definitions of the meaning of component terms reflect the normal use of the words.
3. If lexical definitions reflect normal use, then answering conceptual questions should rely on statistical surveys such as Gallup polls.
4. Therefore, answering conceptual questions should rely on statistical surveys.
5. However, answering conceptual questions should not rely on statistical surveys.

Thus, answering conceptual questions does not involve the study of lexical definitions.

This is a simple reductio and is clearly valid. Lines 2 and 3 imply line 4 via modus ponens. This, however, leads to a contradiction with an additional premise 5, which is accepted by Turing in passing without much discussion. He only states that such attitude would be “dangerous.” Note that contemporary experimental philosophers who rely on statistical surveys and similar empirical methods might also explicitly embrace a premise that implies Line 4 or something similar. It might also seem that if Turing’s assumption is that the study of lexical definitions is not useful for answering conceptual questions, then also, by the same token, the computational methods of distributional semantics are equally dangerous. Nonetheless, to avoid contradiction between lines 4 and 5, Turing apparently suggests that we should simply drop the premise stated on line 1. This then is the indirect argument against relying on lexical definitions in the study of answers to conceptual questions.

What would be the danger of relying on statistical surveys to answer such philosophical questions? Plausibly, Turing, along with philosophers of the ordinary language of his time, was well aware of the fact that usage patterns change over time, and that “reading off” philosophical views from particular linguistic usage is close to making a simple fallacy of
confusing “is” with “ought”. A description of a usage pattern in language does not suffice to justify the general claim that this pattern should be followed, even if uttered in suitable contexts and taken for face value. For example, from the fact that there are many (fossilized) expressions such as “The sun sets”, which might be taken, at face value, to express a commitment to a geocentric theory of universe, it does not follow that the theory is widely assumed to be true or endorsed by any competent speaker uttering such an expression. At the same time, common usage patterns are pieces of evidence used by linguists to formulate prescriptive rules; when prescriptive rules clash with usage, in particular educated usage, they come under heavy criticism (see e.g., (Pullum 2010)).

Turing’s argument against relying on statistical surveys is but one in an ongoing debate between descriptivists and revisionists about philosophical concepts. Instead of understanding and analyzing our existing conceptual machinery, revisionists propose revising them. At the turn of 21st century, Simon Blackburn dubbed the project of assessing and ameliorating our concepts “conceptual engineering” (Blackburn 1999). As is well known, Turing proposes to revise both the question and the way one could answer it: through assessing the machine performance in what we now know call “the Turing test”, we could at least provide a sufficient condition for being intelligent.

Here, my presentation of the Turing test will be kept very brief (for extended discussions, see (Proudfoot 2005; łupkowski 2010; Turing 2004; Copeland 2003; Castelfranchi 2013)). In short, what Descartes thought to be unavailable for any sort of mechanism, viz. communication in natural language, is considered by Turing to be the hallmark of intelligence. In Discourse on the Method, Descartes contended that no mechanism could ever “use words, or put together other signs, as we do in order to declare our thoughts to others” (Descartes 1985, 1:140). Turning the tables, Turing considers natural language use to be sufficient for intelligence. This is the core idea is that there could be sufficient similarity of how machines could declare their thoughts to others, as compared to human beings. The possibility that there be such similarity is the very point of his version of the imitation game, in which the judge is asked to determine whether they converse with a machine or a human being.

Thus, the goal of the imitation game is to test whether a machine might imitate human conversational ability to the extent that it would be indistinguishable from a human being. Being indistinguishable puts the machine into the same class of abstraction, showing that it is equivalent to the human being under this respect, which is deemed by Descartes—and Turing—sufficient for intelligence.

Note that this equivalence obtains only with respect to features of observable behavior. The machine need not provide the same answers, given the same question as a human being, thus it is not considered a weakly equivalent simulation of human behavior, to use Fodor’s

---

1 Some critics of ordinary language philosophy ascribe this fallacy to this research tradition as a whole (Gellner 1959). However, this seems to be insufficiently charitable. Mere frequency or typicality of usage is not sufficient for ordinary language philosophers to justify its correctness. In fact, ordinary language philosophers were much more sophisticated in justifying their normative claims, for example, about category mistakes involved in typical usage patterns, see e.g., (Ryle 1949).
later terminology (Fodor 1968). The process that leads to producing verbal behavior need not be in any way equivalent to the one that occurs in human beings, so it is not strongly equivalent.\(^2\) In other words, the equivalence in question is far less demanding than the ones implied in the subsequent simulation research in psychology.

Turing’s version of the imitation game, adapted from a party game, serves a function of what Dennett calls an “intuition pump” (Dennett 2013): it provides a way to perform a thought experiment to engineer a new concept of intelligence as applied to machines. Following Dennett, one may analyze Turing’s test as a machine for “pumping” intuitions that depend on the machine settings.

The first set of settings ensures that the machine is not easily uncovered by requiring that the communication be purely textual, without the presence of biological bodies, with a proper speed of communication, that the machine be not obliged to declare its true identity, etc. Note that some of these settings could be varied without influencing the result of the test if computational technologies would allow the machine, for example, to speak English and parse spoken English. The essential thing is that the playing ground be even.

The second set of settings is related to the contents of the conversion: it should not be constrained or limited, which could easily make the test too easy. Moreover, the comparison should be made between the machine and a typical adult speaker. The success of Parry in simulating a paranoid patient (Colby 1975) did not constitute a milestone in the development of general artificial intelligence.

Finally, the third set of settings is related to the judges, who should focus on the conversation alone.

As an intuition pump, the Turing test is supposed to help us revise the notion of intelligence we adopt to make it applicable to machine behavior, at least in selected circumstances, just because of the features of machine behavior. In the subsequent literature, Turing’s move has been repeatedly described as offering an operationalist or behaviorist definition of thinking (French 2000; Block 1995). However, as Jack Copeland argues, “[t]here is no textual evidence to support this interpretation of Turing” (Turing 2004, 435). Indeed, Turing explicitly said that he did not want to “give a definition of thinking” (Turing 2004, 494). The goal of this instance of conceptual engineering was to provide a novel understanding of intelligence, by unpacking what it might mean in practice that a machine is actually conversant in natural language. By indicating that there is a machine that passes a test, we could disprove Descartes’ dictum that thinking could not be understood in purely mechanistic terms. Thus, the test’s aim is to deal with the overall project of determining whether thinking (or intelligence) can be mechanized.

\(^2\) A number of critics, starting from Claude Shannon and John McCarthy (Turing 2004, 437), through Stanislaw Lem (Lem 1974), to Ned Block (Block 1981), objected that the process could be quite “nonintelligent”, while still producing intelligent-like conversations. The plausibility of the idea that a “lookup table” is capable of producing and understanding a potentially infinite number of English expressions over 30 minutes of unconstrained conversation is debatable. Our current deep neural networks with hundreds of billions of parameters display remarkably fluent linguistic behavior but remain brittle for common-sense questions.
As a consequence, the test is not supposed to provide any kind of quality metric for intelligent machine systems. Instead, it provides a tool for thinking: it opens up a conceptual possibility that machines could display intelligence. This is exactly why it should be considered an intuition pump: it should lead to the revision of the previously entrenched conviction that thinking cannot be mechanized. The test is supposed to be a proof of concept that it could indeed be the case that machines are intelligent by “declaring their thoughts”. It plays this role even without any machine being able to pass the test, and this could have inspired a certain approach to the study of artificial intelligence: prove that a machine is capable of a certain complex behavior, showing that a certain kind of algorithm is sufficient to produce it, without even trying to mimic actual psychological processes in human beings. Needless to say, this is what the test could have inspired in functionalism in the 1960s and 1970s. This way, it proved its usefulness as a tool of conceptual engineering.

By focusing more on the construction of the intuition pump, we can soon discover that the test makes the notion of intelligence relational. As Proudfoot observes, Turing embraced an “externalist” conception of intelligence: “whether or not machines think is in part determined by social environment, in the form of the interrogator’s responses” (Proudfoot 2005). This is a double-edged sword.

On the one hand, this externalism may undermine intersubjective agreement over a given instance of a test. Judges may disagree. Indeed, substantial disagreement is typical (Aron n.d.). This could be the reason why during a radio debate in 1952, Turing proposed that a jury should decide: “A considerable proportion of a jury, who should not be expert about machines, must be taken in by the pretence” (Turing 2004, 495). This indicates that Turing did not want any preconceptions about machines and computation to take precedence over the conversational dynamics (so this strengthens the set of parameters related to how judges should be selected). As Colby observed, in the original imitation game, whose aim was to distinguish a man mimicking a woman from a real woman, there can be no real experts either, because “there exist no known experts for making judgments along a dimension of womanliness” (Colby 1975). However, this is not a bug but a feature for Turing.

If we were to treat this as a specification of an experimental protocol, then there is a considerable similarity of the roles of the members of the jury and human participants who are asked to classify a certain entity. In contemporary psychology or natural language processing, human participants are asked to provide judgments that are usually limited to a forced choice among several options. These judgments are used in coding the structure ascribed to videos, images, gestures, written or spoken linguistic expressions, etc. Currently, crowd-sourced annotation tasks are usually used to produce datasets for machine learning purposes. In contrast, however, to the Turing test, the choices made by human annotators are usually constrained theoretically by detailed instructions provided by experts. This means that experts are actually responsible for designing such instructions, while judgments are made independently, usually by at least a pair of human beings working independently (to avoid amplifying their individual bias), and then made consistent in various ways. For example, one way to resolve the disagreement is to appeal to a supervising annotator. Another is to hold a discussion between annotators until they resolve the disagreement.
themselves. None of this is proposed for the Turing test – this is partly because the judgments are not used downstream for further processing. However, this is also because the test is supposed to demonstrate that the machine under the test has the conversational capacity that corresponds to a common-sense judgment. After all, one should not expect intuition pumps to provide precise and generalizable performance metrics.

However, there is a price to pay: passing the test, in contrast to other tasks in natural language processing, is not a tangible goal, because it is virtually impossible to break down the task into its components and provide a metric of how good a given model performs. While many language-related capacities, such as translation, are difficult to evaluate automatically, mostly because various human translators would produce vastly different translations given the same original input document, there are some measures that could be used to see whether there is some progress or not (see e.g., Koehn 2010 for a discussion of the BLEU score). Not so for the Turing test. This explains why the Turing test has remained outside the purview of mainstream NLP.

On the other hand, Turing’s approach seems to align well with the conceptual changes inspired by the development of artificial intelligence. As many have noticed, we tend to consider machine performance less intelligent when we become acquainted with it (McCorduck 2004, 204). Douglas Hofstadter credited Larry Tessler as the author of the Theorem that expressed this effect succinctly: “Al is whatever hasn't been done yet.” (Hofstadter 1979, 601) The modern notion of artificial intelligence seems indeed to be “externalist”, or “emotional”, as Turing would have it:

The extent to which we regard something as behaving in an intelligent manner is determined as much by our own state of mind and training as by the properties of the object under consideration. If we are able to explain and predict its behaviour or if there seems to be little underlying plan, we have little temptation to imagine intelligence. With the same object therefore it is possible that one man would consider it as intelligent and another would not; the second man would have found out the rules of its behaviour (Turing 2004, 431).

However, just because people seem to consider themselves experts on machines or algorithms they frequently use (such as the web search engine, which is arguably an instance of fairly complex artificial intelligence), they no longer consider them particularly intelligent. While this is a source of frustration among developers of artificial intelligence, Turing’s original position seems to be: just bite the bullet.

Thus, this conception of intelligence in computing machinery is not only highly interactive and externalist, but also prone to changes over time, similar to any other common-sense conception. Nonetheless, that seems to be the goal of this instance of Turing’s conceptual engineering.

---

3 Turing speculated that a machine would pass his test in a hundred years from 1952, so the situation may look different in thirty years’ time.
Let us now contrast the Turing test with his previous groundbreaking proposal to understand computability in terms of universal machines (Turing 1937). “Computability” used to be an informal concept, made more precise thanks to the new mathematical notion that formalized it, which is now known as the Turing machine. In this case, Turing proposed a definition, a precising definition to be exact, to make the original notion of computation less vague. The precising definition cannot be defended as true (or rejected as false) because, as a non-lexical definition, it is not truth-evaluable. Turing, however, did justify the introduction of the new notion of computability with his simple thought experiments that analyzed what a human computer could do with paper and pencil, finite number of instructions, etc. Thus, Turing also relied on intuition pumps in his 1937 paper but his goal was different. He wanted to defend a particular innovative formalization.

“Intelligence” or “thinking” are informal concepts, like “computability” before 1937, but Turing did not propose to formalize them. One could hypothesize that in contrast to “computability” in the 1930s, which was arguably vague (and understudied), “intelligence” in the 1940s or 1950s did not even refer to computing machines. Thus, Turing would have to propose a theoretical stipulative definition of “thinking” or “intelligence” in formal terms, if he were to follow his previous footsteps. But he clearly wasn’t ready to offer a formal definition and he did not propose a novel, complete mathematical framework. He had genuinely new insights into the role of learning, randomness, organization, and situatedness of intelligent systems in their social environments, which preceded developments in computer science sometimes by many decades, but these did not constitute a comprehensive theory of intelligence. In summary, the role of thought experiments related to what might be accomplished with paper and pencil in his 1937 paper diverged from what was supposed to be accomplished in his 1950 paper: conceptual engineering was aimed at formalization in the first case, while it played a more preliminary motivating role in the latter.

There are, however, two questions that remain unclear in light of what has already been said. First, why did Turing stress the notion of “intelligence” rather than “thought”, “thinking” or “the mind”? It is difficult to find an outright answer in Turing’s writings. One traditional way to provide an answer would be to compare Turing’s reliance of the term with what was happening at the time in British philosophy, cybernetics, and everyday debates. For example, Gilbert Ryle stressed the notion of “intelligence” (or “intelligent” behavior) in his 1949 book, (Ryle 1949). However, Ryle worked at Oxford, and was not a member of the Ratio club, which was a hub of and springboard for the growing cybernetic movement in the UK. There is some evidence that prominent proponents of cybernetics in the UK, such as W. Grey Walter or Donald MacKay, relied on the notion of “intelligent action” rather than “thinking” as well already in the late 1940s (MacKay 1969; Walter 1953). However, the latter one also wrote a paper that answered the very same question that the Turing test was designed for (MacKay 1951). MacKay framed them as a series of interrelated questions, such as the following:

(i) Can an artefact be made to show the behavioural characteristics of an organism?
(ii) How closely in principle could the behaviour of such an artificial organism parallel that of a human mind? (MacKay 1951, 105)

However, the title of MacKay’s paper mentions “mindlike behavior”, not “intelligence. Thus, it remains unclear why Turing relies on the notion of “intelligence” rather than “mindlike behavior” or “thinking”.

Additionally, it remains unclear why Turing did not consider his test to provide a sufficient condition of intelligence. He could have understood it as a working definition to be adjusted in the future. While his denial of the intention to define the notion is univocal, a number of readers were strongly impressed that his paper actually provides a definition, and an operational one. Operational definitions, even if operationalism as a philosophical doctrine about theoretical terms is unviable, do provide clear pointers to other researchers taking up experimental issues. Why not treat the imitation game test as an operational definition, after all?

To answer these two questions, which seem to be unanswered by close reading, we shall turn to computational linguistics.

3. Implicit semantic relations

In this section, we study the interpretational questions about the Turing test from the perspective of natural language processing. The written oeuvre of Alan Turing remains rather small. For the purposes of the current study, a corpus of his published and unpublished work was compiled, including all his work included in (Turing 2004) but also additional work contained in (Turing 1992) (i.e., “Proposal for Development in the Mathematics Division of an Automatic Computing Engine (ACE)”, “Lecture to the London Mathematical Society on 20 February”, “Checking a large routine”) as well as unpublished work on morphogenesis from (Turing, Saunders, and Turing 1992) and some parts of transcripts on Turing’s Enigma book. The resulting corpus has 227,757 tokens (185,052 words).

For topic analysis, which gives an overview of the subject matter in the corpus, we used Lingo4G, developed by CarrotSearch. The algorithm details are proprietary but they go beyond the usual topic modeling (by including some additional heuristics). While topic modeling should not be treated as reflecting any “objective” topical structure of text, it can

---

4 While the writings in (Turing 2004) were already digitalized and proofread, other work had to go through OCR. For this purpose, tesseract OCR engine was used (in the default LSTM setting). Unfortunately, the official website of the Cambridge Turing Archive is down as of (17th January 2022), previously available at www.turingarchive.org. This makes crucial Turing online archives no longer accessible. Unfortunately, these also did not include full transcripts of Turing’s correspondence (OCR for typewritten or handwritten text remains very noisy), such as those found in 2017. For this reason, the corpus does not cover all remaining correspondence of Alan Turing.

5 Because the number of documents in the corpus, understood as separate publications, reports, or letters, is fairly low (below 30), original documents were split into individual paragraphs and processed this way to uncover the underlying topical structure in a fine-grained fashion. It must be stressed that the results should be treated as heuristic rather than “more objective” than those provided by close reading of Turing (see also (Allen and Murdock 2022)).
still provide some insights into its conceptual structure. The analysis was performed on the paragraph level because of data scarcity.

Figure 1 provides the diagram (a treemap) for the topics that can be found using the keyword specified as a wildcard: “intelligen*”. These were clustered together to show closely related topics. At a glance, one can see that Turing inquired into the possibility of whether machines could display intelligence, while still making serious mistakes, playing chess (even if that would be a bad game). One particularly large cluster of topics is related to the notion of the unorganized machine, clearly hypothesized to be a model of (infant) human cortex, human intelligence, and “occasional wrong result”. This analysis suggests that one leitmotif in Turing’s thinking is fallibility related to intelligence, although not necessarily when thinking about human mathematicians.

At the same time, the same diagram also shows interest in the technical means for achieving human-level intelligence, such as having large memory available – clustered together with “genuine intelligence”. This suggests that Turing might have endorsed something like the physical symbol system hypothesis put forward by Allen Newell, who claimed “the hypothesis is that humans are instances of physical symbol systems, and, by virtue of this, mind enters into the physical universe” (Newell 1980, 136). Physical symbol systems are essentially equivalent to universal Turing machines. While not actually having unbounded memory, they only display “powerful and efficient intelligence” if they have “sufficiently open memory” (Newell 1980, 161).

There are also some unclustered topics and some are clearly artifacts of the analysis (such as “9th term”, “added 18”, or “added 21”). This is typical of topic modeling, which usually yields multiple topics, some of which do not provide any suggestive insight. At the same time, it should be noted that this type of modeling is only suggestive of further interpretation (cf. (Allen and Murdock 2022)).
Having sketched the whole topic landscape of Turing’s writing, let us now turn to more detailed insights we can gain by relying on distributional semantics. For the purposes of semantic analysis, the corpus was processed using SketchEngine (www.sketchengine.eu), which is a state-of-the-art corpus analysis software (Kilgarriff et al. 2014).

One interesting feature of SketchEngine is to extract the similarity of meanings of terms in a corpus, based on distributional relations inherent in it. It is therefore fairly easy to study the semantic relations between notions used by Turing. They can be visualized using diagrams (see below Figures 2-5), which contain the term of interest in the center, while the distance from the center depicts the degree of semantic similarity.

It should be noted that similarity is inferred from the resemblance among words embedding the terms of interest appear. To take a simple example, suppose a given corpus contains two statements:

(i) Correlation does not imply causality.
(ii) Correlation does not imply causation.

The software then should infer that “causality” and “causation” are semantically similar because they are embedded in similar statements, comprised of very similar terms in similar grammatical structures. Of course, the size of the corpus increases the reliability of such inference but even for relatively small samples, the results can be useful for further interpretation.

Let us then study terms that are semantically close to “intelligence” first.
Figure 2: Terms semantically close to “intelligence”. Font and circle sizes correspond to the term frequency in the corpus, while distances reflect semantic similarity.

Now, let us complement this with a grammatical variant, “intelligent”, in Figure 3.
Figure 3: The visualization of terms semantically similar to “intelligent” in Turing’s writings.

Before we draw any conclusions, let us now compare this with semantic fields of “mind” (Figure 4) and “thinking” (Figure 5).
Figure 4: This diagram depicts terms semantically close to the term “mind” in Turing’s corpus.
What seems evident is that Turing’s terms “intelligence” and “intelligent” are not related explicitly to “thinking” or “mind”. By looking at Figures 2 and 3, one can easily spot a connection between universal computing machinery and intelligence, but also between randomness, making mistakes, and unorganized machines (this is what we saw in Figure 1 as well). Clearly, what this reflects is Turing’s hypothesis that learning is crucial to intelligence, and that learning can proceed efficiently in a (partially) random fashion. In contrast, “mind” is more related to “knowledge”, but also to “test” (obviously, the imitation game test). Thus, there is a latent connection between having a mind and being intelligent. This is because the imitation game is supposed to play a decisive role in substantiating claims about machine intelligence.

Interestingly, for Turing, the terms “brain” and “mathematician” are equally close to “mind”, but the term “child” is closer. This suggests that even though Turing is best known to have analyzed the human mind as performing calculations (in his famous paper that introduced the concept of the universal machine), he did consider the child’s mind even more important. Even though he would say that it’s inessential that the brain “has the consistency
of cold porridge” (Turing 2004, 477), and although the brain is all but mentioned in his 1950 essay that introduced the idea of the Turing test, he did consider the issue of how the brain contributes to the mind as well. This can be easily seen if we compare the adjectival collocations for “brain” and “mind”, see Figure 6. As this figure shows, Turing used the terms “mind” and “brain” interchangeably when talking about adults and human beings. He would also use the terms “electronic brain” and “mechanical brain” (but critically, see (Turing 2004, 484)). These terms were made popular in the British press after the war (see (Proudfoot and Copeland 2019)).

Figure 6: A comparison of adjectival predicates that accompany the terms “mind” and “brain”. To the left, we find the terms associated mostly with “mind”, to the right – those associated with “brain”.

Figure 4 also reveals a clear connection between thinking and symbolic notation: Turing did consider notation and symbols important, and it is not a coincidence that the symbolic approach to artificial intelligence was also inspired by his work. Good old-fashioned artificial intelligence can also claim descent from Turing’s work.

However, all this does not yet answer our initial question: why would Turing talk of intelligence and not mind, when considering computing machinery? What the corpus analyses show is that the notion of intelligence is much more technical for Turing than the notion of “mind”, which is used mostly in a nontechnical fashion, without any attempts to make it more precise or related to notions such as “learning” or “memory”. This may suggest that the choice was to use a term that was less entrenched in the philosophical tradition of the mind-body problem. Even though “the mind” has a relatively recent origin in modern philosophy, introduced by John Locke in English but still unavailable in direct translation in many other languages (such as French or German), it immediately raises philosophical
concerns such as “asking what could be known with certainty even if the empirical case was granted” (to cite Allen Newell commenting on the philosophical reception of Turing’s 1950 paper (Newell 1973, 47)), instead of inspiring further theoretical and practical exploration. As many of his contemporaries, such as Ryle and MacKay, Turing preferred to focus on intelligent action.

Surely, computational linguistics does not offer groundbreaking new insight into Turing’s choice of terminology. However, it does show that the usages of terms such as “mind” and “intelligence” in his writing are different but not vastly different (for example, the child’s mind is mentioned as a possible target of simulation in his 1950 paper). This means that it is wrong to assume that Turing makes a clear-cut distinction between “mind” and “intelligence”. All in all, this seems to suggest that Turing would probably join his contemporaries in using the term “artificial intelligence” in 1956, rather than “artificial mind”, although he would not make a big fuss about it. Both notions are interconnected in his work.

Let us now turn to the issue of why Turing did not treat his imitation game as the operational definition of intelligence. After all, as many other operational definitions, Turing’s imitation game does suggest a sufficient condition for something to be a definiens – an intelligent entity capable of verbal conversation. Let us then see how Turing would use the term “definition”, by looking at related terms (see Figure 6).
There are two terms that come closest to “definition”: “test” and “example”. This shows two things. First, Turing’s readers who assumed that his 1950 paper offers a definition by offering a test were not hallucinating. There is actual textual evidence that these terms are used in similar situations in Turing’s oeuvre. It is obvious why examples play the role of definitions – these are simply demonstrative definitions. The tests are usually operational definitions. While the notion of “test” is used much less frequently than that of “example” or “definition” (which is represented by the font size), one can, nonetheless, draw a conclusion that Turing’s test is very closely related semantically to a definition. Indeed, there is no reason to suppose that it could not be used as an operational, rough definition of being intelligent. This is the role it plays in Turing’s conceptual engineering.

4. Conclusion

The aim of this paper was to present Turing’s conceptual engineering in two ways: by looking at it through the lens of traditional close reading, and by complementing the close reading by additional insights provided by text mining.

The text mining perspective addresses two questions that remained unresolved by the close reading of Turing’s work. The first issue is: Why Turing did not use the notion of mind, and talked of intelligence instead. The second issue is whether his test could not be used as an
operational definition that provides a sufficient condition for being intelligent. While the size of Turing’s corpus is fairly limited, which makes any quantitative techniques more prone to noise and bias, the analysis suggests some answers to both of these questions. As far as the first one is concerned, the mind/intelligence distinction is not a clear-cut one, and both terms are interrelated (through the notion of learning or education). Turing simply preferred the term “intelligence” when proposing developing machine capacities in solving problems and learning. The second issue can be answered fairly simply: although explicitly Turing denied that his test suggests a definition of thinking, implicit semantic relations suggest otherwise. Actually, the Turing test comes very close to being an operational definition.

The Turing test plays a particular role in how Turing engineers the notion of “intelligence”. It is plausible to suppose that it is easier to engineer notions whose usage patterns are less entrenched in everyday speech and previous theorizing. This makes “intelligence” a much better candidate for conceptual engineering than “mind”. What seems to be the case is that the test’s role is to change our outlook on intelligence in general, so that we could develop a better conceptual and theoretical grasp of what it involves after the test will have been actually passed. However, the test does not offer a rich theoretical understanding, in contrast to what Turing achieved with respect to the notion of “computation”. Given the high standards he had for definitions, one can suspect he would be much more impressed by a theoretical definition that provides more rigorous insight into what intelligence involves. Conceptual engineering must start from what is already available, and the grasp of the notion of intelligence in the 1940s and 1950s was insufficient, by Turing’s standards, to offer a full-fledged theory.

Looking back, it seems that Turing’s attempt to revise our notion of intelligence was a partial success. For some readers, it did open a rich avenue of theoretical and practical exploration, leading to the systematic study of what was soon called “artificial intelligence”. For others, it is as nonconclusive as any other conceptual argument in the debate on the nature of mind as opposed to machine, mechanism vs. biological agency, and so on. However, this is the case for most intuition pumps in philosophy; these do not settle philosophical issues once and for all.

Let me finish with a modest remark. From a meta-theoretical perspective, this paper also shows that while methods of “distant reading” do not offer any silver bullet for difficult philosophical problems, these methods can still provide us with more understanding of our intellectual heritage. Nonetheless, while they are useful for qualitative analysis, their reliability increases with the quantity of text.

Herein, there are serious technical obstacles for computational methods in the service of intellectual history. For example, while a huge number of historical materials were scanned and made publicly available, their availability, even for research purposes, is insufficient. For example, the British Newspaper Archive offers merely a search engine but bulk downloads, detailed textual analysis, and similar functions are not accessible for copyright reasons. Moreover, handwritten manuscripts, typed correspondence and notes all remain very difficult for optical character recognition; instead, they usually must be manually transcribed, which makes the cost of such operations fairly prohibitive at a larger scale.
Ideally, having a corpus of all Turing’s writing, historical newspapers, technical reports, and academic writing in related disciplines, could provide more insight into his work. It is not unreasonable to expect that these resources will become available in the future.

Hopefully, Turing would be pleased that algorithms can offer some insights into his own thinking. While these would not pass his test, they bridge the gap between humanities and artificial intelligence. It is still the job of the human being to interpret the result of the analysis.

References


Dennett, D. C. 2013. Intuition Pumps and Other Tools for Thinking.


http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:Artificial+Intelligence+and+the+Concept+of+Mind#0.


