Gino Roncaglia

Computer che copiano: test di Turing, web corpora e filtraggio collaborativo.

Preprint, versione maggio 2014; l'articolo è in corso di stampa negli Atti del convegno dell'Accademia dei Lincei "Per il centenario di Alan Turing", a cura di Tito Orlandi.

Nell'articolo *Computing Machinery and Intelligence* – uno dei testi chiave nella storia dell'intelligenza artificiale, pubblicato su Mind nel 1950[1] – Alan Turing propone quello che sarebbe diventato famoso come il test di Turing: una procedura pratica per stabilire se un computer sia in grado di manifestare un'intelligenza linguistica di fatto indistinguibile da quella umana. Tradizionalmente, il test di Turing (o le sue possibili varianti) presuppongono che la macchina sottoposta a test sia una, ben individuabile, e – fatta eccezione per l'interazione con l'esaminatore – lavori in maniera autonoma. Turing non discute esplicitamente questi aspetti, ma il paragone con il gioco dell'imitazione, in cui tre persone ben individuate (l'esaminatore, l'uomo, la donna) svolgono ciascuna il proprio compito senza aiuti esterni, suggerisce chiaramente che anche nel caso del test di Turing il computer sottoposto al test debba fare affidamento esclusivo sulla propria programmazione e sui dati di cui dispone.

Del resto, c'è un passo specifico dell'articolo di Turing – alla fine del § 5 – che risulta da questo punto di vista assolutamente inequivoco:

> It was suggested tentatively that the question, "Can machines think?" should be replaced by "Are there imaginable digital computers which would do well in the imitation game?" If we wish we can make this superficially more general and ask "Are there discrete-state machines which would do well?" But in view of the universality property we see that either of these questions is equivalent to this, "Let us fix our attention on one particular digital computer C. Is it true that by modifying this computer to have an adequate storage, suitably increasing its speed of action, and providing it with an appropriate programme, C can be made to play satisfactorily the part of A in the imitation game, the part of B being taken by a man?"[2]

Dato che i computer digitali possono replicare perfettamente qualunque macchina a stati discreti, sostiene Turing, la domanda se vi siano computer digitali capaci di superare il test include la domanda se vi siano macchine a stati discreti capaci di superarlo, e tutte e due le domande equivalgono a chiedersi se vi sia *un particolare computer digitale C* che, adeguatamente

[1] A. M. Turing, *Computing Machinery and Intelligence*, in «Mind» n. 49, 1950, pp. 433-460. Il testo dell'articolo di Turing è disponibile in rete in diversi siti, incluso quello della rivista «Mind», che ne permette la consultazione in una digitalizzazione anastatica dell'originale: http://mind.oxfordjournals.org/content/LIX/236/433.full.pdf+html. Questa e tutte le altre risorse web citate in questo articolo sono state verificate l'ultima volta nel luglio 2013.

[2] *Ivi*, p. 442.

programmato e fornito delle necessarie risorse di memoria, sia in grado di ingannare l'esaminatore umano.

Il computer che affronta il test di Turing, insomma, assomiglia un po' a uno studente che affronta un esame forte solo della sua preparazione, senza poter chiedere aiuti all'esterno, senza poter copiare o barare: una situazione che ben corrisponde a una fase di sviluppo dell'informatica in cui le potenzialità rappresentate dalla telematica e dalle reti non erano ancora percepite.

I primi tentativi di costruire programmi 'intelligenti', in grado – se non di superare – almeno di cimentarsi con il test di Turing o con sue versioni 'settoriali', sono stati direttamente influenzati da questo modello. L'esempio più chiaro è probabilmente rappresentato dai 'chatbot': agenti software nei quali – come nel test di Turing – l'interazione uomo-macchina è esclusivamente linguistica. L'interazione con un chatbot rappresenta senza dubbio la situazione più vicina a quella immaginata da Turing, e la programmazione di un chatbot è in genere guidata dall'idea di ingannare l'interlocutore umano facendogli credere di conversare con una persona, o comunque con un'entità in grado di sostenere una conversazione intelligente attraverso l'uso competente del linguaggio naturale (non a caso ci si riferisce spesso ai chatbot con l'acronimo ACE, 'Artificial Conversational Entity'). Ma l'evoluzione dei chatbot fra gli anni '60 e oggi mostra un progressivo allontanamento dall'idea della totale autosufficienza degli agenti software.

I chatbot 'classici', a partire da Eliza[3] e Parry[4], possono includere nella conversazione anche le risposte o passi delle risposte dell'interlocutore umano (e lo fanno assai spesso), ma non le usano per modificare la propria programmazione o il database linguistico e lessicale utilizzato. Possiamo classificare questi casi come quelli più strettamente aderenti al modello proposto da Turing.

Molti fra i Chatbot di seconda generazione sono invece in grado di apprendere dall'interazione con l'utente: così, ad esempio, Jabberwaky[5] and Cleverbot[6], entrambi programmati dal ricercatore britannico Rollo Carpenter, possono riutilizzare frasi, lemmi e risposte fornite da un utente all'interno dell'interazione con un altro utente, e permettono – volendo – un intervento diretto dell'utente nella programmazione, attraverso la creazione di script che corrispondono a 'personalità' diverse. Capacità in parte analoghe sono possedute da A.L.I.C.E. (Artificial Linguistic Internet Computer Entity), uno dei chatbot di maggior successo: fra il 2000 e il 2004 ha vinto tre

[3] ELIZA è un po' il capostipite dei chatbot, e ha influenzato fortemente tutto il lavoro successivo in questo campo. Programmato fra il 1964 e il 1966 da Joseph Weizenbaum, era utilizzato – in associazione con lo script DOCTOR – per simulare uno psicoterapeuta di scuola Rogeriana. Cf. Joseph Weizenbaum, *ELIZA—A Computer Program For the Study of Natural Language Communication Between Man And Machine*, in «Communications of the ACM» n. 9 (1), January 1966, pp. 36–45 (una digitalizzazione anastatica dell'articolo è disponibile in rete all'indirizzo http://www.cse.buffalo.edu/~rapaport/572/S02/weizenbaum.eliza.1966.pdf). Il programma originale ha conosciuto nel tempo una infinità di variazioni e modifiche ad opera di numerosi ricercatori, e il contesto della conversazione psicoterapeutica rimane uno di quelli più popolari nel mondo dei chatbot.

[4] PARRY è un chatbot programmato fra il 1970 e il 1972 dallo psichiatra Kenneth Mark Colby; simula un paziente schizofrenico paranoide. Cf. Kenneth Mark Colby, Sylvia Weber, Franklin Dennis Hilf, *Artificial Paranoia*, in «Artificial Intelligence» 2 (1971) pp. 1-25 (una digitalizzazione anastatica dell'articolo è disponibile in rete all'indirizzo http://www.csee.umbc.edu/courses/graduate/CMSC671/fall12/resources/colby_71.pdf), e Kenneth Mark Colby, *Artificial Paranoia; A Computer Simulation of Paranoid Processes*, Oxford, Pergamon Press, 1975.

[5] Il sito di riferimento per Jabberwacky è http://www.jabberwacky.com/.

[6] Il sito di riferimento per Cleverbot è http://www.cleverbot.com/.

volte il Loebner Prize, assegnato annualmente al chatbot dalle prestazioni migliori in una situazione analoga a quella del test di Turing[7]. In questi casi, il programma è in grado di allargare progressivamente il proprio database linguistico e lessicale, costruendo una sorta di 'corpus in progress' utilizzato poi per selezionare le proprie risposte.

L'idea di utilizzare non solo l'interazione con gli utenti ma anche la rete come strumento per allargare il corpus linguistico a disposizione di un chatbot rappresenta una evoluzione abbastanza naturale della ricerca in questo settore. Già all'inizio degli anni '80, questa strada è stata sperimentata da Bruce Ellis e Robert Pike nella programmazione di Mark V. Shaney, un agente software in grado di inviare messaggi al gruppo Usenet net.singles costruendone il testo sulla base dei messaggi inviati allo stesso newsgroup da altri utenti[8]. Più che un chatbot, tuttavia, Mark V. Shaney è un esempio di software per la generazione automatica di testi: un'altra tipologia di programmi che tende a basarsi sempre più spesso sull'uso di corpora testuali raccolti in rete.

Negli ultimi anni, anche i programmatori di chatbot veri e propri si sono progressivamente resi conto che l'allargamento – possibilmente massiccio e automatizzato – dei corpora di riferimento costituisce la strategia migliore per il miglioramento delle capacità dei loro programmi[9]. E la ricchezza della rete in termini di contenuti linguistici riutilizzabili ne fa evidentemente uno strumento prezioso proprio per questo tipo di allargamento. ALICE 2.0, la nuova versione di A.L.I.C.E. annunciata nel 2013, è ad esempio in grado di raccogliere informazioni interagendo con alcuni servizi di rete, inclusi altri bot. "These external sources" – è spiegato nel post che annuncia il nuovo software – "provide up-to-date cultural references and information about current events"[10]. Come vedremo in seguito, si tratta di una linea di sviluppo che porta direttamente dal mondo dei chatbot a quello degli 'assistenti virtuali' come Siri o Google Now, presenti in molti smartphone di ultima generazione.

Se analizziamo lo sviluppo che abbiamo sommariamente cercato di ricostruire nel campo dei chatbot, ci rendiamo conto che dalla situazione ideale di totale autosufficienza del programma immaginata da Turing per il suo test ci siamo progressivamente spostati verso sistemi capaci di imparare dalla propria esperienza di interazione con gli utenti, e da questi verso sistemi in grado di utilizzare, in varie forme, corpora testuali esterni di grandi dimensioni. Corpora che possono essere selezionati e consultati in rete, anche durante la stessa interazione con l'utente, al fine di offrire

[7] Il sito di riferimento per A.L.I.C.E. è il blog dedicato al progetto dal suo programmatore, Richard S. Wallace: http://www.alicebot.org/, al cui interno si veda in particolare l'articolo *The Anatomy of Alice*, http://www.alicebot.org/anatomy.html. Anche nel caso di A.L.I.C.E. alla capacità di apprendimento si affianca la possibilità di personalizzazione: in questo caso lo strumento usato è AIML (Artificial Intelligence Mark-up Language), un linguaggio di marcatura XML che consente all'utente sia la programmazione di chatbot del tutto diversi, sia la possibilità di estendere le competenze linguistiche del programma originario aggiungendovi la capacità di riconoscimento e risposta a pattern testuali assenti nella programmazione iniziale. Tre dei primi quattro chatbot nel Loebner Prize 2013 sono programmati usando AIML (cf. http://alicebot.blogspot.it/2013/06/loebner-prize-2013-three-of-four.html). Per una discussione su natura e limiti dei 'contest' come il Loebner prize si veda Huma Shah e Kevin Warwick, *From the Buzzing in Turing's Head to Machine Intelligence Contests*, paper presentato a TCIT 2010: Towards a Comprehensive Intelligence Test - Reconsidering the Turing Test for the 21st Century, 29 Mar-1 Apr 2010, De Montfort University, Leicester, UK, in rete all'indirizzo http://centaur.reading.ac.uk/17371/.

[8] Per alcune informazioni di base si veda http://en.wikipedia.org/wiki/Mark_V_Shaney.

[9] Cf. Bayan Abu Shawar e Eric Steven Atwell, *Using corpora in machine-learning chatbot systems*, in «International Journal of Corpus Linguistics», Volume 10, Number 4, 2005, pp. 489-516.

[10] Cf. http://alicebot.blogspot.it/2013/04/introducing-alice-20.html.

risposte sempre più appropriate e significative. Se il primo di questi sviluppi – la capacità di apprendere dall'esperienza – può essere comunque considerato compatibile con il modello di Turing (dato che in ogni interazione data il computer rappresenta comunque una entità autonoma e autosufficiente)[11], l'idea che *durante l'interazione* il computer possa cercare aiuti esterni finalizzati a migliorare la propria prestazione, ricercando in rete e copiando comportamenti linguistici adottati da esseri umani davanti a domande o affermazioni 'simili' a quelle incontrate durante il test, non poteva essere presente in questa forma all'epoca di Turing. Si potrebbe certo ricondurre anche questo caso al modello originale, postulando che dell'insieme di risorse e di dati a disposizione della macchina faccia parte l'intero web nello stato in cui si trova al momento dell'interazione, ma si tratta evidentemente di un allargamento non di poco conto del concetto di 'macchina' ipotizzato da Turing. In un certo senso, dunque, il computer sottoposto al test di Turing sta imparando a 'barare': quando deve reagire a una frase dell'utente, o rispondere a una domanda, può farlo non solo in base alle conoscenze di cui già dispone, ma anche sulla base di conoscenze e modelli di comportamento linguistico ricercati appositamente in rete, che possono essere copiati e 'incollati' nell'interazione.

E' importante peraltro osservare che quello dei chatbot non è certo l'unico campo di applicazione dell'intelligenza artificiale in cui è percepibile una evoluzione di questo tipo. Un altro esempio classico è rappresentato dai giochi.

Indubbiamente una programmazione 'classica', strettamente logica e autosufficiente, ha portato a risultati assai significativi nel caso di giochi come gli scacchi, tradizionalmente considerati uno dei banchi di prova più adatti a misurare le capacità di analisi e scelta di un computer in un contesto suscettibile di una descrizione puramente formale. Molti ricorderanno la scena classica del film *2012 Odissea nello spazio* in cui il computer HAL 9000 gioca a scacchi con Dave, ovviamente vincendo la sfida: un episodio non casuale, se si considera che fra i consulenti di Stanley Kubrick per la realizzazione del film vi era Marvin Minsky. E la vittoria di Deep Blue su Kasparov nella sfida del maggio 1997 rappresenta in fondo per l'intelligenza artificiale classica uno dei pochi esempi di promesse mantenute, di sviluppi previsti che hanno poi trovato conferma in realizzazioni effettive. Nel contempo, il tipo di programmazione di Deep Blue era perfettamente compatibile con la sostanziale autosufficienza della macchina ipotizzata da Turing nel concepire il suo test: certo, il programma includeva una larga base dati di partite e situazioni effettivamente giocate, utilizzato in particolare per le aperture e le chiusure, e il programma era in grado di 'imparare' anche dalla propria esperienza di gioco; ma – nonostante le accuse mosse all'epoca da Kasparov proprio relativamente a questo punto – per vincere contro l'avversario umano non aveva bisogno di 'barare' attraverso aggiustamenti *in itinere* della programmazione o delle proprie basi dati[12].

[11] Del resto l'idea della necessità di migliorare progressivamente le competenze del computer attraverso una forma di addestramento simile a quello al quale è sottoposto uno studente da parte del docente era ben presente in Turing già nel saggio del 1948 *Intelligent machinery* (in B.J. Copeland, *The essential Turing: The ideas that gave birth to the computer age. Oxford University Press*, Oxford 2004, pp. 395-432; si veda ad es. p. 422, in cui Turing parla ella necessità di addestrare il computer attraverso un processo che funzioni "mimicking education") e torna nelle interviste alla BBC del 1951 (*Intelligent Machinery, a Heretical Theory*, *Ivi* pp. 465-475).

[12] Sulla sfida fra Kasparov e Deep Blue – alla quale grazie a un invito dell'IBM ho avuto la fortuna di poter assistere di persona per conto della trasmissione di RAI Educational *Medi@mente* – e sulla programmazione di Deep

Nel caso di giochi più difficilmente formalizzabili degli scacchi, tuttavia, la situazione cambia. Ad esempio, la programmazione dei 'bot' impegnati in giochi multiutente on-line (MMORPG e simili) ha portato progressivamente allo sviluppo di agenti software che non solo imparano dall'esperienza di gioco ma 'assorbono' e imitano il comportamento dei giocatori umani che incontrano[13]. Bot di questo tipo hanno superato con successo apposite versioni settoriali del test di Turing relativamente a diverse tipologie di giochi, dai MMORPG ai giochi basati su simulazioni di guida (il caso dei MMORPG è tuttavia il più interessante, dato che prevede interazioni complesse fra gli utenti, come contrattazioni e scambi economici, interazione linguistica sia fra i personaggi sia fra i giocatori, esplorazione di territori, scelta di strategie di combattimento, ecc.).

Un altro esempio interessante che – almeno nella sua applicazione più famosa – può essere collegato al campo dei giochi è rappresentato da Watson, il supercomputer dell'IBM divenuto famoso per aver sconfitto i propri avversari umani nel popolare quiz televisivo statunitense *Jeopardy*. Watson – proprio come i giocatori umani – doveva rispondere a domande poste usando il linguaggio naturale. L'aspetto per noi più rilevante è la metodologia usata per la costruzione della base di dati utilizzata da Watson: il corpus di riferimento del programma era infatti costituito quasi interamente da informazioni tratte dalla rete: oltre 200 milioni di pagine web, incluso l'intero contenuto di Wikipedia. Anche se Watson non era fisicamente collegato a Internet durante il gioco, la sua programmazione si basava in sostanza sulla capacità non solo di analizzare la struttura della domanda, ma anche di ricavare dalla rete le risposte giuste[14]. E i programmatori di Watson avevano ben chiaro che una interazione diretta e in tempo reale con l'informazione conservata in rete costituisce la strategia migliore per costruire e migliorare progressivamente agenti software di questo tipo: l'assenza di collegamento durante il gioco era legata solo all'esigenza di mettere formalmente il computer e i giocatori umani nelle stesse condizioni, ma ovviamente la porzione di Web presente nella memoria di Watson (circa quattro terabyte di dati testuali) era abbastanza significativa da rendere di fatto poco rilevante la 'disconnessione' di Watson nel momento della competizione.

Va osservata peraltro, in questo e in altri casi di uso di corpora informativi tratti dal web, la crescente importanza dell'uso di dati fortemente strutturati e standardizzati – anche dal punto di

---

Blue si veda Feng-Hsiung Hsu, *Behind Deep Blue: Building the Computer that Defeated the World Chess Champion*, Princeton University Press, Princeton 2002.

[13] Cf. Philip Hingston (ed.), *Believable Bots: Can Computers Play Like People?*, Springer-Verlag, Berlin 2012. Si veda in particolare il cap. 6, *Believable Bot Navigation via Playback of Human Traces*, di Igor V. Karpov, Jacob Schrum e Risto Miikkulainen (pp. 151-170), che analizza il comportamento di diverse tipologie di agenti software "in the context of the BotPrize competition, a Turing-like test where computer game bots compete by attempting to fool human judges into thinking they are just another human player", concludendo che "the method of direct imitation allows the bot to effectively solve several navigation problems while moving in a human-like fashion" (*Ivi*, p. 151).

[14] Per una presentazione accessibile e introduttiva dei principi di programmazione di Watson e per la storia della sua partecipazione a *Jeopardy* si veda Steven Baker, *Final Jeopardy: Man vs. Machine and the Quest to Know Everything*, Houghton Mifflin Harcourt, New York 2011; per una discussione del rapporto fra il test di Turing e la programmazione di Watson si veda Huma Shah, *Turing's misunderstood imitation game and IBM's Watson success*, Invited talk at The 2nd Towards a Comprehensive Intelligence Test (TCIT) - Reconsidering the Turing test for the 21st Century Symposium, AISB 2011 Convention – York, UK; il testo dell'intervento di Shah è disponibile on-line su Researchgate all'indirizzo http://www.researchgate.net/publication/236889246_Invited_talk_Turings_misunderstood_imitation_game_and_IBMs_Watson_success/file/60b7d519f3bf7aac2c.pdf.

vista della semantica e delle ontologie utilizzate – attraverso il ricorso a metodologie ispirate all'idea di semantic web, in particolare nella forma *linked data*. Wikipedia è così ad esempio utilizzata da Watson non nella forma standard ma nella sua versione linked open data, DBpedia[15].

Un ulteriore campo tradizionalmente rilevante per le ricerche in intelligenza artificiale e nel quale è evidente il progressivo passaggio da una strategia che potremmo definire di 'autosufficienza computazionale' a una strategia basata in sostanza sul trovare e copiare esempi appropriati (sempre più spesso tratti dalla rete) è quello della traduzione automatica. Una discussione ragionevolmente completa e approfondita dell'evoluzione di questo settore[16] richiederebbe spazio e competenze ben maggiori di quelle di cui dispongo. Ma basterà ricordare la celebre contrapposizione fra le metodologie 'razionaliste' che avevano segnato i primi anni di sviluppo del settore (basate sull'idea che il requisito fondamentale per un sistema di traduzione automatica fosse un modello computazionale 'forte' del funzionamento del linguaggio) e quelle empiriste (basate sull'uso di corpora, memorie di traduzione e metodologie statistiche) tematizzata fin dal titolo nella quarta TMI Conference tenuta a Montreal nel 1992[17], per cogliere le prime tracce di uno sviluppo che culmina oggi nell'idea del "web as corpus"[18] e dell'estrazione automatica dalla rete di corpora paralleli massivi[19].

Anche grazie a questi sviluppi, i sistemi di traduzione automatica web-based – a partire da Google Translate – hanno migliorato enormemente in pochi anni la loro efficacia. Oggi Google Translate utilizza con successo metodologie di filtraggio collaborativo[20] per far valutare agli stessi

---

[15] Cf. David Ferrucci, Eric Brown, Jennifer Chu-Carroll, James Fan, David Gondek, Aditya A. Kalyanpur, Adam Lally, J. William Murdock, Eric Nyberg, John Prager, Nico Schlaefer, Chris Welty, *Building Watson: An Overview of the DeepQA Project*, in «AI Magazine» 31(3), 2010, pp. 59-79 (l'articolo è disponibile in rete all'indirizzo http://www.aaai.org/ojs/index.php/aimagazine/article/viewArticle/2303), p. 69.

[16] Per una introduzione classica al tema si veda W. John Hutchins e Harold L. Somers, *An Introduction to Machine Translation*, Academic Press, London 1992; per una sintesi più recente, particolarmente attenta all'uso dei metodi statistici, si veda Philipp Koehn, *Statistical Machine Translation*, Cambridge University Press, Cambridge 2010; per una sintesi storica sull'evoluzione della traduzione automatica si veda W. John Hutchins, *Machine Translation Over Fifty Years*, in «Histoire Épistémologie Langage» vol. 23 / 1, 2001, pp. 7-31, disponibile in rete all'indirizzo http://www.persee.fr/web/revues/home/prescript/article/hel_0750-8069_2001_num_23_1_2815.

[17] *Proceedings of the Fourth International Conference on Theoretical and Methodological Issues in Machine Translation: Empiricist vs. Rationalist Methods in MT* – TMI-92, Montreal, Canada 1992. Già nel 1988, nella seconda TMI Conference, Peter Brown dell'IBM aveva attaccato direttamente il paradigma 'razionalista' con la celebre affermazione secondo cui le prestazioni del suo sistema miglioravano sensibilmente ogni volta che licenziava un linguista. Cfr. al riguardo la ricostruzione e la discussione presenti in Harold Somers, *Review Article: Example-Based Machine Translation*, in «Machine Translation» 14, 1999, pp. 113-157.

[18] Cf. «Journal of Computational Linguistics» - *Special issue on web as corpus*, Volume 29 Issue 3, September 2003, e la serie dei "Web as corpus Workshop" organizzati dallo Special Interest Group of the Association for Computational Linguistics (ACL) on Web as Corpus, arrivati nel 2013 all'ottava edizione (http://sigwac.org.uk/).

[19] Cf. ad es. Bill Dolan, Chris Quirk, Chris Brockett, *Unsupervised construction of large paraphrase corpora: exploiting massively parallel news sources*, in COLING '04, Proceedings of the 20th international conference on Computational Linguistics, disponibile in rete partendo dalla pagina http://dl.acm.org/citation.cfm?id=1220406.

[20] L'idea alla base del filtraggio collaborativo è l'uso sistematico delle preferenze degli utenti, associato a meccanismi di profilazione dell'utenza, come strumento per costruire sistemi di raccomandazione. Ma il filtraggio collaborativo può essere usato anche per far emergere automaticamente, attraverso le scelte degli utenti, determinati contenuti, o i risultati più adeguati per una ricerca, o – nel caso che qui ci interessa – le scelte di traduzione considerate più adeguate dal maggior numero di utenti. Per una analisi delle principali tecniche di filtraggio collaborativo si veda Xiaoyuan Su e Taghi M. Khoshgoftaar, *A survey of collaborative filtering techniques*, in «Advances in Artificial Intelligence», January 2009, disponibile on-line partendo dalla pagina http://dl.acm.org/citation.cfm?id=1722966.

utenti la bontà della traduzione proposta, con la possibilità da parte dell'utente di suggerire traduzioni alternative e migliori. E l'azienda di Mountain View non fa mistero del fatto di voler sfruttare anche l'enorme patrimonio di testi digitalizzati nell'ambito del progetto Google Books come base per la creazione di corpora paralleli multilingue utilizzabili nelle traduzioni. Uno strumento come Google translate, che fino a qualche anno fa era guardato dagli addetti ai lavori con assoluto scetticismo, si sta in tal modo trasformando non solo in una risorsa progressivamente più utile ed efficace[21], ma in una palestra di sperimentazione che unisce 'big data' e collaborative filtering, e che affronta le sfide della traduzione automatica affidandosi largamente a una sorta di saccheggio organizzato e sistematico dei molti e diversi corpora offerti dalla rete.

Per concludere questa rassegna, non può mancare almeno un accenno a uno fra gli sviluppi recenti più interessanti dal punto di vista che qui ci interessa: quello rappresentato dagli "assistenti vocali intelligenti" come Siri (sviluppato da Apple per il sistema operativo iOS utilizzato nei propri dispositivi mobili, a partire dall'iPhone) e Google Voice Search (sviluppato da Google per il sistema operativo Android e incorporato in Google Now, l'incarnazione più recente del celebre motore di ricerca). In entrambi i casi si tratta di agenti di ricerca che riprendono, nell'interazione con l'utente, alcune fra le caratteristiche dei chatbot, ma le integrano con la capacità di interagire in tempo reale con la rete, utilizzandola per rispondere alle domande dell'utente; domande che vengono raffinate e talvolta addirittura anticipate grazie all'uso sistematico di servizi di geolocalizzazione, di profilazione dell'utenza e – ancora una volta – di filtraggio collaborativo.

Il risultato è spesso sorprendente; in un articolo dal significativo titolo *The Apple iPhone 4S, Siri, and the Turing Test: the creepiest piece of technology I've seen in a while*, il giornalista scientifico del «Telegraph» Tom Chivers ha scritto:

> I'm not, by any means, suggesting that the iPhone 4S is actually anywhere near that [i.e. passing the Turing test]. But it is (…) the closest of anything I've ever seen; it's the first real-life computer to give me the Uncanny Valley effect just by talking. You can't help but wonder, while watching it, how long it will be before it says, with chilling calm, "I'm sorry Dave. I'm afraid I can't do that".[22]

E' ancora presto per avere valutazioni affidabili sull'impatto che queste tipologie di strumenti potranno avere sul futuro dell'intelligenza artificiale. Quel che appare chiaro, tuttavia, è che negli ultimi anni gli sviluppi della rete hanno radicalmente cambiato le metodologie di lavoro e di programmazione degli agenti software, e in particolare la ricerca legata allo sviluppo di agenti 'intelligenti'. Sempre più spesso, programmi e applicazioni 'intelligenti' sfruttano intensivamente le risorse di rete, selezionando quelle di volta in volta rilevanti sulla base di un insieme di criteri abbastanza differenziati: dagli strumenti di organizzazione semantica basati su *linked data* ad

[21] Per un confronto recente fra diversi sistemi di traduzione on-line, dal quale Google Translate emerge come incontrastato vincitore, si veda ad es. Claire Ellender, Free Online Translators: A Comparative Assessment of www.worldlingo.com, www.freetranslation.com, and www.translate.google.com, in Translation Journal vol. 16 n. 3, July 2012, on line all'indirizzo http://www.bokorlang.com/journal/61freexlation.htm.

[22] Tom Chivers, *The Apple iPhone 4S, Siri, and the Turing Test: the creepiest piece of technology I've seen in a while*, «The Telegraph», 5 ottobre 2011, http://blogs.telegraph.co.uk/news/tomchiversscience/100109144/the-apple-iphone-4s-siri-and-the-turing-test/.

algoritmi di pertinenza e di filtraggio collaborativo, che utilizzano sistematicamente esperienze già acquisite 'simili' a quella che ci si trova ad affrontare per replicare le risposte di maggior successo, senza disdegnare affatto l'idea di 'copiare' risposte di successo reperite in rete.

I passi avanti fatti in pochi anni utilizzando queste strategie sono notevoli: certo, la necessità di includere nella programmazione degli agenti software 'intelligenti' anche una componente logico-deduttiva non è in discussione; ma l'impressione è che tale componente debba essere affiancata da una robusta attività di 'raccolta e imitazione', largamente basata sull'uso delle risorse disponibili in rete. La capacità di selezionare e riutilizzare bene e velocemente contenuti e comportamenti linguistici reperibili in rete sembra insomma rappresentare in molti casi una strategia concretamente più efficace rispetto al tentativo di costruirli autonomamente sulla base di un modello astratto.

English summary

In his famous paper on Computing Machinery and Intelligence, published in 1950, Alan Turing suggests an empirical procedure to determine a computer's ability to exhibit a human-like intelligent behavior: the Turing test.

Traditionally, Turing test (as well as most of its many different variants) presupposes that the computer works in a fully autonomous way – the only exception being its interaction with the examiner. This means that in the test situation, the computer can only use its software and the data already included in its databases, without any external help. This is a somehow 'natural' and well understandable assumption at a time in which the relevance of computer networks was still to be fully understood. Deep Blue's victory against Kasparov in the 1997 chess match offered a good example of a computer working in this way; but Deep Blue might also have been one of the last instances of this approach to artificial intelligence.

In more recent years, the development of networks has radically changed the situation. 'Intelligent' software makes an increasing use of network resources and of external data collected on the fly. From linked data to ranking algorithms, from collaborative filtering to search and recommendation tools, collecting external data concerning situations 'similar' to the one that has to be dealt with, in order to reuse them and to replicate the behaviors that proved to be most effective for other agents, is now a common and successful strategy.

Interesting examples are offered by intelligent vocal assistants such as Siri or Google Voice Search, and by tools for automatic translation based on massive translation memories on the cloud. In such cases, while interacting with the user, the software systematically relies upon the 'help' of relevant network resources, and works by 'copying' successful answers and behaviors found elsewhere, without caring too much for their origins (be they human or artificial).

This strategy seems to have achieved impressive results in just a few years. Apparently, the ability of find and reuse pertinent answers already present on the net is in many cases more effective than trying to 'derive' them from an abstract theoretical model (such as a computational model of the language).

Abstract in italiano

Nell'articolo *Computing Machinery and Intelligence* – uno dei testi chiave nella storia dell'intelligenza artificiale, pubblicato su Mind nel 1950 – Alan Turing propone quello che sarebbe diventato famoso come il test di Turing: una procedura pratica per stabilire se un computer sia in grado di manifestare un'intelligenza linguistica di fatto indistinguibile da quella umana. Tradizionalmente, il test di Turing (o le sue possibili varianti) presuppone che – fatta eccezione per l'interazione con l'esaminatore – la macchina lavori in maniera autonoma. In altri termini, il computer sottoposto al test fa affidamento esclusivo sulla propria programmazione e sui dati di cui dispone, e non può ricorrere ad aiuti esterni: una situazione che ben corrisponde a una fase di sviluppo dell'informatica in cui le potenzialità rappresentate alla telematica e dalle reti non erano ancora percepite. La vittoria di Deep Blue su Kasparov, nella sfida del maggio 1997, rappresenta un buon esempio di questa concezione, e ne costituisce nel contempo un po' il canto del cigno.

Negli ultimi anni, gli sviluppi degli strumenti di rete hanno radicalmente cambiato la situazione. Sempre più spesso, programmi e applicazioni 'intelligenti' sfruttano intensivamente le risorse di rete, selezionando quelle di volta in volta rilevanti sulla base di un insieme di criteri abbastanza differenziati: dagli strumenti di classificazione semantica basati su *linked data,* ad algoritmi di pertinenza e di filtraggio collaborativo, che utilizzano sistematicamente esperienze già acquisite 'simili' a quella che ci si trova ad affrontare, per replicarne le risposte di maggior successo.

Esempi interessanti sono offerti dagli "assistenti vocali intelligenti" come Siri e Google Voice Search, e dagli strumenti di traduzione automatica basati sull'uso massiccio di memorie di traduzione gestite 'on the cloud'. In entrambi i casi, il programma interagisce con l'utente cercando sistematicamente aiuto dalla rete. E, se ne ha modo, 'copia' risposte di successo reperite in rete, senza preoccuparsi troppo dell'origine (umana o artificiale) della risposta che viene utilizzata.

I passi avanti fatti in pochi anni utilizzando queste strategie sono notevoli: l'impressione è che la capacità di selezionare e 'copiare' bene e velocemente le risposte migliori reperibili in rete rappresenti in molti casi una strategia concretamente più efficace rispetto al tentativo di costruirle sulla base di un modello astratto (ad esempio, un particolare modello di funzionamento del linguaggio).