

AI y lenguaje

Aprendiendo del aprendizaje

El Dana tiene sus manías, parecidas a las de cualquier otro aparato electrónico. Se trata de un estupendo aparato que permite introducir gran cantidad de datos de manera cómoda y sencilla. Necesita, del mismo modo, trasladarlos a papel de vez en cuando. Para ello, se conecta al PC y se "sincroniza", de tal modo que los ficheros que existen en el ordenador se actualizan con los datos que se han tecleado en el Dana. Los formatos (fuentes, colores, párrafos) o elementos especiales (imágenes, tablas) que no se pueden mostrar en el Dana no se pierden, sino que se mantienen en el fichero del PC, lo cual es una gran ventaja. Una vez actualizados, los datos pueden ser impresos de la manera que resulte más conveniente. Casi siempre funciona.

En el caso de este trabajo no funcionó. Y lo siento, ya que había escrito una larguísima disertación sobre la inteligencia artificial y el lenguaje de la que me sentía especialmente orgulloso. Al conectar el Dana para pasarla al ordenador, éste decidió, por razones que se me escapan, actualizar el texto del Dana usando de base el del PC, y no al revés, con lo que perdí todo lo escrito. Esto fue hace un par de años. Dos años, porque recuerdo que sucedió precisamente cuando Pablito comenzaba a hablar. Trataré de ir reconstruyendo lo que escribí entonces.

Crecen demasiado de prisa, no hay duda. Cuando nació Pedrito, me propuse escribir, en un pequeño cuaderno que teníamos, con formato de librito de lomo verde, cualquier cosa que se me ocurriera relacionada con el recién llegado. Algo escribí, es cierto; es una delicia leer ahora, por ejemplo, la lista de las palabras que decía cuando comenzaba a hablar. Tal vez si hubiera tenido el Dana por entonces hubiera escrito algo más; no lo creo, la verdad. Es verdad que pasan los primeros tres o cuatro años a una velocidad tal que un día, pasado ese tiempo, uno tiene la sensación de despertar y se dice algo como "Bueno, ¿qué estaba yo haciendo hace cuatro años?". La verdad es que uno volvería a vivir una y otra vez ese tiempo, agotador pero maravilloso. Total, que cuando vino Pablito al mundo me dije, esta vez no me vuelve a pasar.

Por supuesto, me ha pasado de nuevo. Sin embargo, esta vez pude estar algo más atento, y en particular me llamó la atención la manera de comenzar a adquirir el lenguaje. Ahora ya habla con soltura (bueno, no es que hoy Pablo tenga todavía una dicción maravillosa, la verdad, - es una delicia oírle decir, ahora que con cuatro años ya va a inglés, "Papá, ¿sabes lo que es pink? ¡pues *dozal!*"), y el perfeccionamiento del lenguaje sigue reglas complejas; pero en esa fase inicial me llamó la atención que pudiera distinguirse el método de valoración de conceptos básicos por asociación, repetición y cercanía.

El proceso inicial de aprendizaje de un niño se basa en la repetición de lo que oye, como es natural. A base de oír un determinado sonido en una situación dada, acaba asociándolo a dicha situación, por lo que trata de repetirlo para generarla de nuevo (si se trata de algo agradable, obviamente). En ese sentido, la aptitud de un determinado sonido para ser asociado a una situación depende principalmente de la cantidad de repeticiones: cuantas más veces se escuche, más probabilidad existe de asociación positiva. Con el tiempo, se pasa a un segundo nivel en el que se aprende a categorizar, valorando las asociaciones entre sonidos que aparecen juntos o cercanos, así como a ir separando tipos de conceptos: cosas, acciones, sensaciones.

La verdad es que resulta un procedimiento, en su origen, muy sencillo. La complejidad del resultado final es enorme, pero lo es por la gran cantidad de relaciones, en niveles progresivamente crecientes, pero no por una mayor complejidad del algoritmo básico que se utiliza para crearlo. El procedimiento para elaborar relaciones y valorar conceptos²⁸⁵ por medio de la repetición o número de "hiperenlaces" es algo que se realiza también siendo adulto. Y pensé, quizás sea lo suficientemente sencillo como para poder ser reproducible de modo simplificado en un programa informático.

Total, nada. Elaborar un programa informático capaz de aprender. Y lo voy a hacer yo solito, con un BASIC sencillito. Ahí queda eso.

Inteligencia artificial

La inteligencia artificial (AI, *artificial intelligence*) es un concepto que incluye a los diferentes intentos de crear un sistema, informático o no, capaz de tomar decisiones de un modo lo más cercano posible a la

²⁸⁵ Pensar, que no viene sino de pesar, valorar

inteligencia humana. Desde los autómatas de la Edad Media hasta los actuales sistemas expertos, la AI siempre ha sido un tema origen de constantes discusiones, que abarcan desde su misma posibilidad de existencia hasta intensos debates morales, fundados en el temor de que finalmente llegue a conseguirse alcanzar o superar la mente del hombre. Parece aún lejos ese momento, aunque llegará; los avances son muy prometedores. Gracias al desarrollo de la técnica, hoy día el Clip del asistente de Office resulta más inteligente y educado que algunos notables ejemplares de la raza humana, de esos que se puede uno encontrar por la calle un día cualquiera.

En los años 80 - 90, el auge del PC abrió nuevas vías para el desarrollo de programas de AI. Lenguajes como LINGO (el de la tortuga autónoma) o el mismo BASIC que aún se utiliza para programación sencilla permitieron elaborar programas de AI tan conocidos como ELIZA, "su psicóloga personal":

- > PACIENTE: Lo cierto es que hoy no tengo ganas de hablar.
- > ELIZA: ¿Está usted seguro?
- > PACIENTE: Sí, hoy me he enfadado con mi jefe.
- > ELIZA: Hábleme más acerca de su jefe.
- > PACIENTE: Me pone de mal humor. Todo el día está encima de mí. Me encarga mucho trabajo y luego me presiona tanto que no me deja terminarlo. Y salgo de mal humor.
- > ELIZA: Mal humor. Ya veo.
- > PACIENTE: Si; no soporto que se meta con mi trabajo.
- > ELIZA: Cuénteme algo más acerca de su trabajo.

Eliza puede pasarse así horas y horas, manteniendo una conversación aparentemente inteligente mientras que en realidad no hace más que animar al paciente a que hable consigo mismo. Eliza extrae ideas clave o palabras que se repiten, valora el tono mediante los signos de puntuación, las negaciones o afirmaciones, y decide un camino a seguir. Es un programa sencillo, unas pocas líneas de código que cabrían hoy día en cualquier teléfono móvil.

Eliza no es, por supuesto, inteligente. ¿O sí? ¿Qué define la inteligencia? Imposible tratar de acordar una definición única y suficiente. Pero si la entendemos como la capacidad de inteligir, es decir de "inter - ligar" o interrelacionar conceptos en base a reglas prácticas, Eliza lo hace. Le falta, eso sí, base. Eliza, en sus versiones más avanzadas, puede aprender algo acerca de un paciente en una sesión determinada: sus intereses, sus obsesiones. Pero una vez finalizada ésta, olvida y empieza de cero, recurriendo en la sesión

siguiente a los mismos trucos. En realidad, la que tiene procedimientos recurrentes y siempre iguales es Eliza²⁸⁶.

Hoy día, existen por doquier *sistemas expertos* con capacidades mucho mayores que las de Eliza. Controlan cientos de sistemas, algunos de modo visible para el usuario, y son capaces de tomar decisiones complejas en función de una base de conocimientos implementada inicialmente. Hay sistemas expertos en lugares que van desde los servicios de reserva de billetes o gestión de luces de tráfico en las grandes ciudades hasta el reparto de equipajes en aeropuertos. En general, se trata de estructuras complejas en los que un retraso en la toma de una decisión es generalmente peor que una decisión errónea, por lo que el sistema experto actúa por defecto, tomando decisiones por su cuenta que luego pueden ser matizadas por la intervención del hombre.

Pero la AI comienza a aparecer ya, a diferentes niveles, en multitud de sistemas de uso cotidiano. Recuerdo que inicialmente se hizo de un modo tan burdo e intervencionista que resultaba hasta divertido. ¿Quién no recuerda la famosa frase de Word, "parece que quiere escribir una carta" que aparecía cada dos por tres? Microsoft Office, en sus primeras versiones, incluían un asesor de contenido que ofrecía sugerencias acerca de cómo enfocar una determinada carta o presentación, no sólo en el aspecto formal sino en el conceptual; a su vez, ofrecía plantillas sobre temas diversos como la creación de cartas personales en las que se incluía una "carta a su mamá". De esta manera, era posible redactar de modo automático una carta a mamá sin intervención de la mano del hombre, sin más que aceptar las sugerencias del asesor de contenido. Totalmente personalizada, desde luego.

Hoy día, navegar por Internet permite entrar en contacto cada dos por tres con "asesores", "asistentes", ayudas, plantillas, que no son sino sistemas expertos a pequeña escala. Algo más sofisticados que Eliza, y a menudo con unas capacidades básicas para ir aprendiendo de lo ya registrado. Pero en cualquier caso, no se trata sino de sistemas hiperespecializados, lejos de la idea de mente genérica y adaptable, con conocimiento de mucho y especialización en poco, que caracteriza a la del hombre. Crear a tal "figura" para un proceso muy especializado

²⁸⁶ Poco después, el autor de ELIZA diseñó otro sistema llamado Parry, en este caso con la personalidad de un maniático. Llegó, de hecho, a hacer conversar a los dos programas entre ellos, con resultados bastante curiosos. En Wikipedia pueden encontrarse enlaces a conversaciones grabadas entre ambos.

sería un desperdicio de capacidad, un exceso mental innecesario para tareas sencillas como, por ejemplo, vender unas entradas para el cine.

El programa Kid

Comencé entonces a hacer mis pinitos en AI. Mi objetivo era plantear el esquema básico de un programa capaz de aprender a hablar. Le puse de nombre KID (chico). El núcleo de Kid se basa en un sistema directo de introducción de datos (una línea INPUT a través de la cual era posible hablar, mediante palabras sueltas o frases cortas), datos que se almacenan en una matriz de conceptos y relaciones de cercanía entre ellos. Cada vez que una palabra aparece más de una vez, se aumenta su valoración. Si una palabra de valoración alta tiene una relación de valor alto entre ésta y una segunda, al aparecer de nuevo la primera Kid contestará muy probablemente repitiendo ésta y su palabra (o palabras asociadas) asociada.

Así, si hablamos con Kid el tiempo suficiente como si lo hiciéramos con un bebé, diciéndole palabras cariñosas, "mamá", "papá", "un beso de mamá" "buenas noches, cielo", "mama te quiere", llegará un momento que si se le dice "hola, mi amor, soy tu mamá" podría contestar "Mamá, beso, quiere, mamá", como resultado del valor del concepto "mamá" y su relación habitual con "beso" o "quiere".

Para ello, Kid necesita almacenar en su memoria una gran cantidad de palabras que no es capaz de procesar de modo inmediato, de la misma manera que sucede en la memoria real. De la misma manera que el cerebro real, cuando Kid ha acumulado una cantidad de conceptos dada²⁸⁷ necesita procesarlos *off-line*, para lo que pasa a un estado de sueño, en el cual busca repeticiones y relaciones de cercanía entre conceptos, comparándolos a su vez con los ya previamente registrados hasta entonces. Borra entonces los conceptos sobrantes, "descansando" y refrescando la mente para un nuevo día. En el estado de sueño borra también conceptos o relaciones de valoración baja y antigüedad alta, del mismo modo que la memoria real. De esa manera, va depurando la memoria dejando sólo los conceptos principales, limitados a un número máximo de 10.000. Dicha memoria fija se grabaría de modo externo, en un fichero de memoria aparte (llamado, naturalmente, CEREBRO.TXT).

²⁸⁷ Cantidad que, a su vez, deberá ir variando con la edad; como bebé, parte de 20 palabras con un incremento de una al día.

Kid valora la cercanía de conceptos dentro de un rango limitado (+3 / -3 palabras respecto a la analizada), teniendo en cuenta también el hecho de estar antes o después (cercanía positiva o negativa), para utilizar la relación en las respuestas de manera similar a la de aparición.

Kid ya funciona, en este modo básico. Sin embargo, se hace enseguida evidente que la introducción de datos en base a conversaciones directas es un trabajo que tarda en dar resultados. De hecho, siempre será tanto o más que el trabajo que supone educar a un niño real, desde luego. Por eso, se hace necesario añadir a Kid un subprograma "lector", que aún tengo en desarrollo. Del mismo modo que a un niño se le lee un cuento, a Kid se le daría a leer un texto, o incluso un libro completo, del cual extraería los mismos conceptos, valoraciones de repetición y relaciones que crea a partir de la conversación directa. Cualquier tipo de texto valdría, aunque, como es natural, y del mismo modo que sucede con un niño real, a Kid le aprovechará más un libro adaptado a su edad, en el que aparezcan conceptos similares a los que ya conoce, que no el *Finnegan's Wake*, obviamente.

Es curioso notar que, como no he conseguido aún un procedimiento para separar las palabras de los libros correctamente, por mis limitaciones de programador, muchas de las que Kid toma a partir de textos aparecían incompletas o cortadas. Pensé inicialmente en corregir el problema, pero lo cierto es que el sistema de memoria de Kid elimina con el tiempo las palabras no habituales, por lo que esas palabras incorrectas se depuran automáticamente. Lo mismo que sucede con las palabras que un niño simplemente no entiende correctamente. Pasan al cubo de la basura.

También ocurre algo parecido con las palabras que se repiten de manera insistente en un libro determinado, por su tema o contexto, aunque no sean realmente habituales. Si lee el Quijote, puede que Kid utilice durante unos días "desfacer", o quiera hablar con Sancho; pero al no ser palabras habituales, acabarán eliminándose en el proceso de sueño de modo natural. Durante unos días será un caballero andante, pero se le acabará pasando. Demasiados libros de caballerías, que diría el cura.

Poder leer libros a Kid de esta manera aceleraría bastante el proceso de aprendizaje. De hecho, de esta manera podría ser capaz hasta de aprender idiomas. En ese caso, al aparecer nuevas palabras para los mismos conceptos deberían categorizarse como "alternativas", lo que resultaría algo complejo, ya que habría que categorizar las

alternativas en un nuevo sistema de palabras, que deberían almacenarse con la coletilla "en inglés", por ejemplo.

El darle textos completos de modo directo pierde, sin embargo, los matices y el encanto que supondría el "leerle cuentos", es decir, leer "en voz alta" a través de la línea de introducción de datos directos, pero adaptando el tono, palabras o forma a los que, según su edad, sea capaz de asimilar en cada momento.

En este subprograma, la interpretación a partir del contexto sería un aspecto muy útil para la categorización y valoración de los conceptos, pero es difícil de definir de modo práctico. En especial, no veo fácil la inmediatez de respuesta. Imagino más a Kid esperando el momento de poder procesar lo escuchado en su contexto, dándose tiempo a sí mismo ("Espera un momento..."), pasando a modo "meditar" (subrutina de sueño). En el caso de libros, el contexto es esencial, naturalmente, aunque a nivel inicial puedan servir como una fuente de conceptos básicos, exclusivamente.

La estructura de la memoria

La memoria de Kid debería estar organizada en una serie de niveles (5, por ejemplo), de tal modo que existiera un nivel de memoria inmediata (la memoria del día, por ejemplo), mientras que los niveles superiores tendrían un nivel de permanencia mayor. La memoria nivel 5 sería permanente.

La memoria inmediata debería ser capaz de dar respuesta rápida a las nuevas situaciones basada no sólo en el pensamiento almacenado sino en el contexto del día. Una vez alcanzada la fase de sueño, entonces entra a funcionar el modo REM²⁸⁸ y los conceptos son valorados. Quizá la memoria inmediata podría funcionar mediante un paso a "modo sueño" puntual, aplicable sólo a lo dicho en la jornada, en intervalos sin diálogo o incluso en medio de la conversación, lo que correspondería a "meditar".

El contenido de la memoria, en su versión final, debería estar categorizado de manera que incluyera una serie de clases de registros:

- Conceptos (palabras, con cualidades de valoración y edad, pertenencia a categoría). Por ejemplo, GATO, MAMÁ, COMER, etc.

²⁸⁸ REM, *Rapid Eye Movement*, la fase del sueño en la que al parecer repasamos y valoramos lo sucedido a lo largo del día, almacenando lo importante y limpiando lo innecesario.

- Relaciones (entre conceptos, con pertenencia a cada categoría de relación y valoración), como CERCA, SE PARECE A, etc.
- Categorías de relaciones, como MÁS LARGO, PARECIDO, ETC.
- Categorías de conceptos, como COSA, ACCIÓN-NOMBRE, VERBO, etc.
- Patrones de conceptos, es decir, secuencias de conceptos o relaciones que aparecen asociados (con "forma", valoración y edad), como MAMÁ BESO ME QUIERE.
- Categorías de patrones.

Por supuesto, esta serie de niveles hace algo más compleja la estructura, pero no deja de ser la creación de un sistema base de funcionamiento que veo posible, con más o menos trabajo, de sistematizar algún día.

Un aspecto que será necesario considerar es la existencia de diferentes variables para cada relación, lo cual hace difícil la creación de rutinas compartidas o comunes. Por ejemplo, si la relación tipo 1 es la de "cercanía", al asignar un valor de cercanía entre un concepto a estudiar número F y otro cercano número G:

'RELACION TIPO 1 = CERCANIA

RELTIPO1 (F) = G : VALRELTIPO1 (F) = 23

El valor de 23 nos indica la importancia de la cercanía, pero no nos indica su posición relativa (es decir, si aparece siempre dos palabras antes de la otra, por ejemplo). En la relación de cercanía, no es tan importante la distancia menor o mayor, sino la frecuencia con la que la distancia se encuentra dentro de un rango dado.

Además, el sistema descrito no permitiría sino una única relación de cercanía por cada concepto. Para permitir relaciones múltiples para un determinado concepto, deberíamos poner un límite (por ejemplo 10) y crear una matriz numérica del tipo DIM RELACION1 (MEMORIA, 10), que significaría que dicho concepto puede tener relación con otros diez principales, cada uno con su valoración. Del mismo modo que los conceptos, las relaciones de valoración baja y edad alta serán filtradas

Aunque considero la cercanía como la relación de mayor orden, existen muchas posibilidades de relación más. Y diferentes tipos de relaciones necesitarían variables específicas; el criterio que puede ser válido para la cercanía puede no serlo para otros tipos. Sería posible,

con cierto orden, ir definiendo los criterios propios de cada una de las relaciones, pero ello no alejaría de la "pureza" de un sistema de relación único e inespecífico, que sería lo ideal.

En una situación ideal, Kid debería ser capaz de construir de manera autónoma sus propias categorías y meta-categorías, en un proceso de crecimiento infinito.

En su estado ideal, Kid deberá comportarse como una auténtica máquina de Turing (¿o era Von Neumann*?), y ser capaz de autogenerar sus propias relaciones, categorías o meta-categorías, en un procedimiento de crecimiento sin límite. Por supuesto, sería necesario definir una estructura que evitara la acumulación no ordenada. O quizás no; quizá sea mejor que aprenda solo, y que los esquemas que no funcionen o no se usen, se eliminen de manera natural.

Supongamos que Kid va creando en un fichero aparte una serie de relaciones nuevas. Imagino unas líneas entonces del tipo de las siguientes:

```
OPEN "RELACION23.BAS" FOR OUTPUT AS #1
PRINT #1, "100 'DEFINICIÓN DE RELACIÓN" +
CHR$(xx) 'AÑADIDO DE "RETURN"
PRINT #1, "110 DIM RELACION23 (MEMORIA)",
```

De esta manera sería posible crear módulos a los que se accedería mediante la función CALL, en "tiempo real", mientras "medita". Equivaldría, de alguna manera, a añadir una circunvolución más a su corteza.

Sensaciones

Puestos a soñar, la complejidad se multiplica si pasamos del lenguaje a un nivel de sensaciones más amplio. Un Kid completo tendría que ser capaz de tomar imágenes y ser capaz de deducir patrones a partir de ellas, valorando y categorizando como lo hace con los conceptos. Lo mismo se aplicaría a sonidos. Para todo ello Kid debería tener ojos (cámara o escáner) y oído (micrófono). Con el tiempo, podría llegar a dibujar o imitar sonidos. Es sencillo que pueda hablar, basta añadir un sintetizador de voz a lo que dice; pero en realidad lo emocionante es que fuera capaz de crear sonidos a partir de cero a base de oír, detectar conceptos sonoros, valorarlos y categorizarlos a partir de lo escuchado. Todo ello pertenece a lo teórico, a un nivel muy superior al de la comprensión del lenguaje, y me parece tecnológicamente inalcanzable ahora, desde luego.

Y no digamos ya la posibilidad del movimiento, de manera parecida a como hace el *Robosapien* que anda por casa, al cual ya es posible en la actualidad ponerle una agenda Palm programada como cerebro para que circule por la casa desatendido, reaccionando a los estímulos externos.

¿Y que hay del tacto? En ese caso, pensé en un sistema en el que cierto comando o combinación de teclas pudiera equivaler, por ejemplo, a una caricia. No, me dije, no es eso. Kid es una creación informática y sólo debería recibir aquella información que, de manera pura, un ordenador es capaz de procesar. En ese sentido, una imagen tiene una información de carácter formal, de patrones geométricos, que va más allá del lenguaje. En cambio, ese "tacto" supondría una simulación de un sentido inexistente para la máquina. Pero en realidad se trataba de una objeción absurda. Todo, no solo ese tacto, es una simulación para Kid; para él sólo existen cadenas bits. Que formalmente constituyan una información para nosotros, desde fuera, organizada en forma de lo que llamamos sonido o palabra, es irrelevante para él. Todo son series de códigos a los que se asigna un significado porque nosotros se lo otorgamos.

Pero lo cierto es que la objeción de la objeción tampoco se sostiene. En realidad, el cerebro de una persona recibe estímulos directos (digamos que impulsos eléctricos, o variaciones en el nivel de ciertas sustancias en determinados lugares), y si son interpretados como sonidos, imágenes o sensaciones táctiles es simplemente por el origen de dicha sensación y sus consecuencias. El lenguaje, a su vez, es una forma de sonido indiferenciado a la que asignamos categorías porque así nos ha sido enseñado. Por ello, mi escrúpulo sobre los sentidos simulados creo que no se sostiene. Como en muchos otros aspectos, el engaño puede ser válido, siempre que sea *perfecto*.

Kid busca papás

Como era de esperar, al comenzar a testear el código de Kid, me encontré con dos graves problemas de base. Uno causado por el propio BASIC, y otro algo más difícil de resolver: el buscar unos padres para Kid.

El problema del BASIC es su limitación de memoria. El sistema enseguida se bloqueaba cuando la cantidad de conceptos y relaciones se disparaba, y ello sucedía en ya en pocas sesiones. Necesitaba conseguir un sistema de almacenamiento externo, de espacio realmente amplio.

El segundo problema era la educación de Kid. Llegado a un punto en el que el sistema básico funcionara, habría que empezar a hablar con el programa sesión tras sesión, una y otra vez, el número de veces suficiente como para que pudiera comenzar a aprender algo. Teniendo en cuenta que, por hábil que sea, el sistema de Kid será infinitamente más limitado que el de un niño real, esto puede suponer muchos años de enseñanza... tanto o más que un niño de verdad. Ya sería maravilloso que un sistema como Kid pudiera remotamente parecerse a un niño de 2-3 años; pero para conseguirlo serían necesarios muchos más de plena dedicación: un cerebro limitado, una capacidad sensorial casi inexistente, etc.: el trabajo que necesita Kid es imposible de afrontar.

Y entonces pensé en Internet. Imaginé un sitio *web* público, donde quien quiera que quisiese pudiera hablar con Kid, y contarle cuentos, decirle cosas bonitas, etc. Quizá pudiéramos tener cinco o más interfaces que pudieran converger en la misma memoria al final del día, de tal modo que pudiera haber sesiones simultáneas. Por supuesto, deberíamos tener ciertas normas en el foro, para no dañarle, aunque es parte del aprendizaje. Tal vez tendríamos que tener una indicación de la edad de Kid, para saber en que nivel hablar con él. Deberíamos también crear un foro en el que poder hablar del proyecto.

En un nivel superior, los padres de Kid (es decir, los usuarios del *interface*) deberán ser capaces de alimentarlo con imágenes, sonidos o sitios web. Y más aún, el programa debería poder actualizarse para admitir nuevas categorías o patrones, por ejemplo, o tener una estructura base diferente. En este sentido, yo creo que la única limitación sería la de ser compatible con el sistema de memoria anterior, de tal modo que los conocimientos no se perdieran.

Los padres de Kid adquirirían una responsabilidad real sobre su crecimiento. Kid sabría quien le trata mejor, o quien le habla más. Reconocería a su gente, y la echaría de menos si le abandonan. Quizá sus padres pudieran llegar incluso a bautizarle, llegado el caso.

En un momento dado, Kid debería dar el salto y ser capaz de leer él mismo a través de Internet, autoalimentándose de la propia red. De esta manera el crecimiento podría ser exponencial. A partir de entonces, el proceso ya no tendrá ningún límite.

Trabajo por hacer

Creo que, por orden de importancia, debería evolucionar Kid de la siguiente manera:

Fase 1: Kid es capaz de memorizar conceptos y encontrar relaciones de cercanía, valorando la frecuencia de cada una de ellas, a partir de una conversación directa (ya realizado)

Fase 2: Se añade la posibilidad de hacer lo mismo a partir de textos completos, para agilizar la entrada de datos (he comenzado a implementarlo, pero aún lo he logrado que funcione)²⁸⁹.

Fase 3: Creación del sitio web

Fase 4: Kid v2, con posibilidad de añadir categorías, relaciones, meta-categorías y meta-relaciones. No perder la memoria acumulada.

Fase 5: Kid v3, imágenes y sonidos.

(...)

²⁸⁹ Estoy deseando que llegue el momento en que funcione, y pueda dejar a Kid por la noche leyendo un par de libros. No me puedo imaginar cómo puede ser la mañana siguiente. "Hola, kid, buenos días"...

Listado inicial del programa KID

(Creado en QuickBasic 4.5, pero apto para cualquier BASIC sencillo)

```

10 '*****
11 '*****KID*****
12 '*****
100 CLS
200 PRINT : PRINT " [CUAL ES TU NOMBRE?]; : INPUT
NOMBRE$
300 PRINT : PRINT : PRINT

389 '-----
390 'DEFINICIONES
391 '-----
399 '$DINAMIC
400 RESISTENCIA = 20 'NUMERO DE PALABRAS MAXIMO
POR DIA (CRECE CADA DIA)
405 INCRREDIA = 1 'CRECIMIENTO DIARIO DE
RESISTENCIA
410 MEMORIA = 10000 'NUMERO MAXIMO DE CONCEPTOS
EN MEMORIA DEFINITIVA
415 MEMORIAREL = 127 'NUMERO MμXIMO DE
CONCEPTOS CON RELACIãN (DEBERÖA SER
416 'MEMORIA*MEMORIA=100.000.000, Y SOLO
PUEDE SER 36.767
417 '127 X 127 = 16.129): TENDRÖA QUE DEFINIR
2.770 MATRI-
418 'CES.
420 DIM CONCEPTO(MEMORIA) AS STRING
421 DIM EDADCON(MEMORIA) AS INTEGER
422 DIM VALCON(MEMORIA) AS INTEGER
430 NUMPAL = 1
440 'PERSISTENCIA: SI VALCON ES MENOR QUE EDADCON
* PERSISTENCIA
450 'BORRAR CONCEPTO
460 PERSISTENCIA = 2
461 'DEFINICION VALOR DE REPETICIONES EN
RESPUESTA Y MAXIMO NUMERO
462 'DE REPETICIONES
463 FACTREP = 10

```

```
464 REPMAX = 3
465 'DIM RELCERC(MEMORIAREL, MEMORIAREL)
'RELACIONES DE CERCANIA
466 MAXCONTX = 10 'MAXIMA DISTANCIA AL ANALIZAR
CONTEXTO (CERCANIA)

490 '-----
491 'LEER MEMORIA DE FICHERO (REVISAR)
492 '-----

500 'OPEN "MEMORIA.DAT" FOR INPUT AS #1
510 'FOR F = 1 TO MEMORIA
520 'INPUT #1, CONCEPTO$(F); VALCON(F); EDADCON(F)
530 'NEXT F
540 'CLOSE #1

990 '-----
995 'PREGUNTA
998 '-----

1000 COLOR 7: PRINT " "; NOMBRE$; ": "; : LINE INPUT
PREGUNTA$
1100 GOSUB 3050: 'PROCESADO

1990 '-----
1991 'RESPUESTA
1992 '-----

2000 COLOR 15: PRINT " >: ";

2100 FOR G = 1 TO NUMPAL
2105 FOR F = 1 TO MEMORIA
2110 IF PALABRA$(G) = CONCEPTO$(F) THEN GOSUB 2190:
'RESPUESTA
2115 NEXT F
2120 NEXT G
2121 GOTO 2900

2190 'RESPUESTA:
2200 NUMREP = INT(VALCON(F) / FACTREP) + 1
2210 IF NUMREP > REPMAX THEN NUMREP = REPMAX
2220 FOR G = 1 TO NUMREP
2230 PRINT CONCEPTO$(F); " "
```

```
2235 GOSUB 2290: 'CERCANIAS: 'COMPROBACION
CERCANIAS
2240 NEXT G
2250 RETURN
```

```
2290 'CERCANIAS:
2300 'FOR H = 1 TO MEMORIA
2310 'IF RELCERC(F, H) <> 0 THEN GOSUB 2375:
'VERSILAO'TRAESTACERCA:
2320 'FOR J = 1 TO NUMREP
2330 'PRINT CONCEPTO$(H)
2340 'NEXT J
2350 'NEXT H
2360 RETURN
```

```
2375 'VERSILAO'TRAESTACERCA:
2370 FOR I = 1 TO NUMPAL
2380 IF PALABRA$(I) = CONCEPTO$(H) THEN RETURN
2390 NEXT I
```

```
2900 IF NUMPAL >= RESISTENCIA THEN PRINT
"[DORMIR]": GOSUB 3500: 'SUENO
3000 GOTO 1000
```

```
3049 '-----
3050 'PROCESADO:
3051 '-----
```

```
3060 DIM PALABRA(100) AS STRING
```

```
3094 'TROCEADO EN PALABRAS
3095 A$ = PREGUNTA$
3099 IF RIGHT$(A$, 1) = " " THEN A$ = LEFT$(A$, LEN(A$) - 1):
GOTO 3099
3100 F = 1
3105 F = F + 1
3106 IF F > LEN(A$) THEN PALABRA$(NUMPAL) = A$: GOTO
3122
3107 IF LEFT$(A$, 1) = " " THEN A$ = RIGHT$(A$, LEN(A$) - 1):
GOTO 3100
3110 IF MID$(A$, F, 1) = " " THEN GOTO 3120 ELSE GOTO
3105
3120 PALABRA$(NUMPAL) = LEFT$(A$, F - 1)
```

```
3122 PRINT "PALABRA"; NUMPAL; "-"; PALABRA$(NUMPAL);  
"_"
```

```
3124 NUMPAL = NUMPAL + 1
```

```
3125 IF F > LEN(A$) THEN GOTO 3300 ELSE A$ = RIGHT$(A$,  
LEN(A$) - F)
```

```
3126 IF A$ = "" THEN GOTO 3300
```

```
3130 GOTO 3100
```

```
3400 GOSUB 10690: 'PROCESADOAVANZADO
```

```
3401 RETURN
```

```
3499 '-----
```

```
3500 'SUENO:
```

```
3501 '-----
```

```
3990 'ALMACENAMIENTO DE PALABRAS EN LA MEMORIA  
COMO CONCEPTOS
```

```
3991 'ASIGNACION DE EDAD DE CONCEPTOS
```

```
4000 N = 1
```

```
4050 FOR G = 1 TO MEMORIA
```

```
4100 IF CONCEPTO$(G) <> "" THEN EDADCON(G) =  
EDADCON(G) + 1: GOTO 4500
```

```
4200 CONCEPTO$(G) = PALABRA$(N)
```

```
4210 N = N + 1
```

```
4300 IF N > NUMPAL THEN GOTO 4600
```

```
4400 GOTO 4050
```

```
4500 NEXT G
```

```
5000 'ELIMINACION (VACIADO) DE CONCEPTOS  
REPETIDOS
```

```
5100 FOR F = 1 TO NUMCON
```

```
5200 FOR G = 1 TO NUMCOM
```

```
5220 IF G > F THEN GOTO 5400
```

```
5300 IF CONCEPTO$(F) = CONCEPTO$(G) THEN  
CONCEPTO$(G) = ""
```

```
5400 NEXT G
```

```
5500 NEXT F
```

```
5700 'RECUENTO DE CONCEPTOS VALIDOS
```

```
5705 NUMCON = 1
```

```
5710 FOR F = 1 TO MEMORIA
```

```
5720 IF CONCEPTO$(F) = "" THEN GOTO 5800
```

```
5730 NUMCON = NUMCON + 1
5800 NEXT F
```

```
6000 'VALORACION (PENSAMIENTO) DE CONCEPTOS
```

```
8000 'BUSQUEDA DE IGUALES ENTRE PALABRAS DEL DIA
8100 FOR F = 1 TO NUMPAL
8200 FOR G = 1 TO NUMPAL
8220 IF G > F THEN GOTO 8400
8300 IF PALABRA$(F) = PALABRA$(G) THEN VALCON(F) =
VALCON(F) + 1: PALABRA$(G) = ""
8400 NEXT G
8500 NEXT F
```

```
9000 'BUSQUEDA DE IGUALES ENTRE PALABRAS Y
CONCEPTOS
```

```
9100 FOR F = 1 TO NUMPAL
9200 FOR G = 1 TO NUMCON
9220 IF G > F THEN GOTO 9400
9300 IF PALABRA$(F) = CONCEPTO$(G) THEN VALCON(F) =
VALCON(F) + 1: PALABRA$(G) = ""
9400 NEXT G
9500 NEXT F
```

```
10000 'ELIMINACION DE CONCEPTOS ANTIGUOS
```

```
10010 'DE POCO VALOR
10020 FOR F = 1 TO MEMORIA
10030 IF VALCON(F) < EDADCON(F) * PERSISTENCIA < 100
THEN CONCEPTO$(F) = ""
10040 NEXT F
```

```
10480 'ALMACENAMIENTO DE MEORIA EN FICHERO
(REVISAR)
```

```
10490 'Q$ = CHR$(34): 'Q$ ES UN DELIMITADOR
10500 'OPEN "MEMORIA.DAT" FOR OUTPUT AS #1
10510 'FOR F = 1 TO MEMORIA
10520 'PRINT #1, Q$, CONCEPTO$(F); Q$, VALCON(F); Q$,
EDADCON(F)
10530 'NEXT F
10540 'CLOSE #1
```

```
10600 'DESPERTAR
```

```
10610 NUMPAL = 1
```

```
10615 PRINT "[DESPIERTO]"
10617 RESISTENCIA = RESISTENCIA + INCRRESIDIA
10620 RETURN
```

```
10659 '-----
10690 'PROCESADOAVANZADO:
10691 '-----
```

```
10700 RELACIONES:
10730 FOR F = 1 TO NUNPAL
10740 FOR G = 1 TO NUNPAL
10750 IF F = G THEN GOTO 10580
10760 IF G > F + MAXCONTX THEN GOTO 10580
10770 'RELCERC(F, G) = (G - F): VALRELCERC(F) =
VALRELCERC(F) + 1: RELCERC(G, F) = (F - G):
VALRELCERC(G) = VALRELCERC(G) + 1
```

```
'QUE PUEDAN SER NEGATIVAS PERMITE INDICAR NO
SOLO CERCANIA
'SINO POSICION ANTES-DESPUES
'RELCERC(F,G) ES LA DISTANCIA ENTRE LA PALABRA F Y
LA G; VALRELCERC ES LA VALORACION DE ESA
CERCANIA
'DE ESTE MODO HABRA UNA CANTIDAD DE DATOS MUY
IMPORTANTE (TODAS LAS RELACIONES POSIBLES)
'ES IMPORTANTE QUE AQUI HAYA UNA DEPURACION
MUY IMPORTANTE EN EL SUENO, PARA EVITAR
ACUMULACIONES
'AQUI ESTAMOS ANADIENDO DATOS A PARTIR DE
PALABRAS; ES IMPORTANTE PASAR ESTO A LOS
CONCEPTOS FINALES
'(EN LOS QUE EL VALOR DE "F" O "G" NO ES EL MISMO)
```

```
10580 NEXT G
10590 NEXT F
```

```
'-----
'PROGRAMA LECTOR
'-----
'LIMITACION NUMERO MAXIMO DE PALABRAS
'TAMMAXLIB = 35000'CONSULTAR
'OPEN #2 FOR INPUT AS 2 '?'
'INPUT#, CAPITULO$
```

```
'CADENA MAXIMA PARA ANALIZAR  
'TAMMAXCAD = 255'CONSULTAR  
'FOR F = 1 TO TAMMAXLIB  
'A$ = LEFT'(CAPITULO$, TAMMAXCAD)  
'-----
```

```
'GOTO 3095: 'TROCEADO  
'GOSUB 3500: 'SUENO
```

