Técnicas de reconocimiento del habla: entre la precisión y la velocidad

Speech recognition techniques: precision or speed

Monika Woszczyna

A medida que los sistemas de reconocimiento de voz avanzan, los desarrolladores se enfrentan al dilema de optar entre incrementar la velocidad o la precisión. La autora ha experimentado con nuevos algoritmos, orientados a conseguir que ambas características dejen de ser opuestas y excluyentes. A continuación, reproducimos algunos extractos de su tesis que repasan las principales técnicas de reconocimiento de voz y algunas soluciones para mejorar su funcionamiento.

As voice recognition systems improve, developers are facing the dilemma of having to choose between increasing speed or precision. The author has experimented with new algorithms, with the aim of finding a way so these two features are no longer opposed and mutually exclusive. The article includes various excerpts from the author's doctoral thesis summarizing major voice recognition techniques and provides some solutions aimed at improving the final output.

De forma progresiva, la complejidad de los sistemas de reconocimiento de voz se ha ido incrementando junto con su precisión, por desgracia a expensas de la velocidad de éste. La razón de este desarrollo está asociada a un criterio único en la evaluación para sistemas de reconocimiento de voz: la exactitud de las palabras en los bancos de pruebas. Con este paradigma, la velocidad de reconocimiento sólo es relevante cuando limita el número de experimentos que pueden llevarse a cabo para elevar la precisión de las palabras. Para conseguir una punta de rendimiento, los reconocedores normales utilizados en tales evaluaciones tardan varios cientos de segundos en procesar un simple segundo de voz introducida en los ordenadores actuales.
Sin embargo, para la mayoría de las aplicaciones del mundo real, la velocidad de reconocimiento es tan importante como la precisión en el reconocimiento. Un sistema de dictado que invierta cuatro días en procesar media hora de habla no será un buen producto. Un usuario no aceptará que una interfaz de base de datos tarde quince minutos en responder a una consulta de cuatro segundos. Argüir que la velocidad carece de importancia porque cada nueva generación de programas es más veloz es válido sólo en parte. Por un lado, los sistemas tienden a hacerse más complejos en una progresión superior a la del aumento de velocidad de los ordenadores. Por otro, se supone que un sistema de reconocimiento de voz aplicado como interfaz para la introducción de datos no debería devorar el 100% de los recursos disponibles.

Concepto de la solución

En muchos casos, los algoritmos y técnicas más rápidos incluyen aproximaciones que reducen la precisión del reconocimiento del sistema. La aceleración y la pérdida de definición resultantes dependen del número de parámetros contemplados.
Comprender por qué y en qué medida la precisión se ve influida por la introducción de nuevos algoritmos en el proceso de reconocimiento aporta valiosas pistas para futuras investigaciones. Para guiar esta investigación, se analizó la distribución del tiempo más allá de las subtareas del proceso de reconocimiento. Se estudiaron en detalle todas las subtareas que contribuían al esfuerzo computacional de forma significativa. Tras explotar las mejoras más prometedoras encontradas en la literatura con el objetivo de reducir más el tiempo dedicado a las correspondientes subtareas se desarrollaron e introdujeron nuevos algoritmos. Utilizando los nuevos descubrimientos de la investigación presentados en esta tesis se construyó un sistema de dictado con un vocabulario de 65 000 palabras. Este sistema de dictado funciona en tiempo real, 200 veces más rápido que el reconocedor de evaluación original en su estado primigenio.

Fundamentos del reconocimiento de voz

En el reconocimiento de voz, la pregunta crucial a responder mediante experimentos es: ¿mejorará la aproximación sugerida la nueva información del reconocimiento? Es importante que la información utilizada para verificar una teoría no haya sido utilizada de ninguna forma cuando se construye el reconocedor. En general, existen tres grupos de datos utilizados para construir un sistema de reconocimiento: información del entrenamiento, información de las pruebas e información de la evaluación.
La información de entrenamiento se utiliza para construir el reconocedor y ajustar sus muchos parámetros. La cantidad de información requerida depende de la aproximación al reconocimiento. En la mayoría de sistemas de reconocimiento, un incremento del entrenamiento implica mejores resultados de reconocimiento.
La información de pruebas se emplea para evaluar nuevos algoritmos durante la fase de desarrollo del reconocimiento. Dado que muchas decisiones se toman basándose en esta información (por ejemplo, si hay que utilizar el algoritmo A o B en el sistema), ésta se «contamina». Las decisiones no pueden ser independientes del grupo de prueba y el rendimiento resultante puede ser superior en la información de las pruebas que en la que se encuentra oculta por completo.
La información de evaluación tiene como objetivo el asesoramiento final sobre el sistema, por lo que debería tratarse de información oculta. Esto significa que ninguno de los parámetros del sistema se ha ajustado a dicha información y que no se ha tomado ninguna decisión basándose en los resultados de la misma. La cantidad de información de pruebas y de evaluación influye en la fiabilidad de los resultados. Por ello, hay que esperar una desviación máxima de los resultados originales cuando se pruebe con un grupo de pruebas distinto.

Medir el rendimiento

Reconocimiento de la precisión
La *tasa de palabras correctas* en el reconocimiento de habla continua se obtiene del siguiente modo:[1]

Tasa de palabras correctas = 100 · número de palabras reconocidas correctamente / número de palabras emitidas

Las palabras que se insertan por error no se incluyen cuando se computa la tasa de palabras correctas. A mayor introducción de errores en una frase, más fácil resulta que éstos correspondan a una palabra que efectivamente se pronunció, aunque la utilidad práctica del *output* de reconocimiento sea muy bajo. Por esta razón, para medir la calidad del reconocimiento se usan más a menudo la *tasa de errores de palabras* y la *precisión de palabras*:[2]

Tasa de errores de palabras = 100 · (sustituciones + elisiones + inserciones) / número de palabras emitidas

La *precisión en las palabras* se define como:[3]

100 − tasa de errores de palabras

Intervalos de fiabilidad
La tasa de palabras correctas se obtiene de la media entre un grupo de dos valores cerrados: cada palabra reconocida puede ser correcta (1) con una probabilidad (p) o incorrecta (0) con una probabilidad (1 − p).
Según Bamberg y Baur (1996), los límites (Vo/u) del intervalo de fiabilidad de tal distribución se computan del siguiente modo:[4]

$$V_{o/u} = \frac{2nX + c^2 \pm c\sqrt{4nX(1-X) + c^2}}{2(n + c^2)}$$

Aquí *n* es el número de palabras del grupo de prueba y *X* equivale a la tasa de palabras correctas. Asimismo, *c* = 1,96 para un intervalo de fiabilidad del 95 %, dado *n* > 100 puede hallarse mediante búsqueda en tabla o por integración numérica.
Para decidir si el rendimiento de dos reconocimientos que trabajan sobre la misma información de prueba difieren significativamente, puede analizarse la distribución de errores en el grupo de prueba (Gilick y Coz, 1989). Para cada grabación en el grupo de prueba se calcula la diferencia de Precisión de Palabras entre ambas pruebas. Se asume entonces que la desviación de 0 de la media sobre estos valores viene dada por fluctuaciones aleatorias. A no ser que esta hipótesis pueda descartarse, la diferencia entre los resultados no será significativa.

Velocidad de reconocimiento
La velocidad de reconocimiento se define como el tiempo de reconocimiento necesario en segundos por cada segundo de habla introducido. Este tiempo puede medirse en segundos o en segundos de CPU, es decir, el período de tiempo que el proceso de reconocimiento tiene el uso exclusivo de la unidad de proceso de un ordenador (con un sistema operativo multitarea tipo UNIX).
Dado que el tiempo empleado para dar acceso a la red o para cargar páginas de memoria virtual del disco no se cuenta en segundos de CPU, hay que tener cuidado con que los nuevos algoritmos no incrementen el tiempo real de reconocimiento mientras reducen el tiempo de CPU. La ventaja de emplear segundos de CPU es que son inherentemente independientes de la cantidad de memoria central disponible y de la carga de trabajo de otros procesos sobre la CPU, lo que facilita las comparaciones entre pruebas.
Obviamente, el tiempo de reconocimiento depende del ordenador y del compilador utilizado para el experimento. En adelante, el tiempo de reconocimiento se da siempre en segundos de CPU. A no ser que el proceso de reconocimiento esté dividido explícitamente en varios pasos, el tiempo de reconocimiento cubre siempre todas las fases del proceso de reconocimiento, incluidos todos los archivos de entrada/salida y el preproceso.
Como unidad de tiempo de reconocimiento utilizaremos el factor de tiempo real (RT). 2RT significa que, para reconocer una grabación el ordenador necesita el doble de tiempo que empleó el emisor en expresarla. Idealmente, el tiempo transcurrido durante la grabación se utiliza también para el reconocimiento de voz. Si un sistema en tiempo real es *pipelined*, el retraso entre el final de la introducción de datos y el reconocimiento puede llegar a ser despreciable.

Intervalos de fiabilidad
Para computar la fiabilidad de los intervalos en el tiempo de reconocimiento, las pruebas se subdividieron en un número de segmentos lo suficientemente grande. Si todos los errores de un segmento se asumen como independientes de los errores del resto de segmentos, la diferencia y la media *X* de la velocidad de reconocimiento pueden computarse asumiendo una distribución normal. Los límites $V_{o/u}$ del intervalo de fiabilidad se computan así:[5]

$$V_{o/u} = X \pm \sigma c / \sqrt{n}$$

En este caso *n* equivale al número de grabaciones y *c* = 1,96 para *n* > 100, dado un intervalo de fiabilidad del 95 %.

Modelación de la señal del habla
El objetivo del reconocimiento del habla es encontrar la secuencia de palabras *W* para una señal dada *A*. Esta es la secuencia de palabras por la cual es la mayor:[6]

$$P(W \mid A) = P(W)\, P(A|W) / P(A)$$

P*(W)* da la probabilidad de una determinada secuencia de palabras y es independiente de la introducción de la señal. Por ejemplo, *but does it* (pero lo hace) es mucho más probable que se dé en

cualquier oración que la secuencia *but doses it* (pero lo dosifica).
P*(W | A)* es la probabilidad de observar la señal *A* dado que la secuencia real de palabras es *W*.
P*(A)* es la probabilidad de la señal grabada. Una vez que la señal ha sido grabada, P*(A)* es igual para todas las secuencias de palabras que puedan haber sido dichas. Así pues, P*(W | A)* es más grande para la secuencia de palabras para la que el producto P*(W)* P*(A|W)* es mayor.
Las siguientes secciones presentan metodologías para computar P*(W)* y P*(W | A)*:

- Los *modelos de lenguaje* muestran un simple algoritmo para estimar P*(W)*.
- Las *pruebas y vectores de funciones* explican cómo la señal del habla *A* se hace accesible al ordenador.
- Los *fonemas y diccionarios de pronunciación* describen cómo las palabras en *W* pueden dividirse en unidades características del habla más pequeñas (fonemas) para simplificar el modelado de P*(W | A)*, la palabra.
- Los *modelos Markov escondidos* introducen el concepto de representación de los fonemas como una cadena de estados S_j que emiten vectores de características I_i como un modelo generativo para P *(W | A)*.
- Las *probabilidades de observación* otorgan un método para estimar la probable densidad $f(I_j \mid S_i)$ de un simple vector de función producido por un único estado.
- El *algoritmo de avance* presenta un algoritmo para computar de hecho P*(W | A)* para una secuencia de palabras dada *W* empleando los métodos introducidos en las secciones previas.
- El *algoritmo Viterbi* ofrece un algoritmo más rápido que puede utilizarse para aproximar P*(W | A)* a la mejor secuencia de estado a través de una secuencia de palabras *W* dada.

Reconocimiento del habla continua
Empleando el algoritmo de avance se puede determinar para cada secuencia de palabras *W* con qué probabilidad la señal *A* fue producida por la cadena Markov correspondiente. Sin embargo, no resulta práctico enumerar todas las secuencias de palabras posibles para encontrar la secuencia que maximiza *P(W | A)*. Si la grabación está segmentada en secciones A_i que contienen exactamente una palabra *W*, $P(A_i \mid W)$ puede ser computada por cada sección para encontrar la secuencia de palabras que mejor encaje en *A*.[7]

Búsqueda en tiempo síncrono
Un enfoque habitual para encontrar la mejor hipótesis de frase de una muestra de habla continua es el *algoritmo de deformación dinámica del tiempo en un único paso (One-Stage-Dynamic-Time-Warping)*, recogido por Sakoe y Chiba (1978), Sakoe (1979) y Ney (1984). Se trata de un algoritmo Viterbi generalizado que combina la segmentación con el proceso de reconocimiento. En el seno de la cadena Markov, y para una palabra, el algoritmo es capaz de encontrar la probabilidad por el mejor camino a través de la palabra empleando el algoritmo Viterbi. Con todo, para cada *frame* existen transiciones adicionales posibles desde todos los finales de palabras a todos los estados de inicio de palabra. Mientras en el algoritmo Viterbi normal el único precedente legal de un estado de inicio de palabra es ese mismo estado de inicio de palabra, todos los estados de final de palabra se consideran ahora posibles predecesores.[8]
Dado que las multiplicaciones requeridas pueden ser reemplazadas por sumas en el espacio logarítmico es habitual utilizar el logaritmo negativo de las probabilidades acumulativas. En consecuencia, una puntuación alta implica una baja probabilidad y una puntuación baja, una alta probabilidad. Para reconstruir la mejor secuencia de palabras, la única información requerida es qué palabra es el mejor precedente al final de la palabra en curso. Así, para cada final de palabra, se almacena la siguiente información: cuál era el mejor antecesor de la palabra y en qué *frame* se dio la transición desde el predecesor a la palabra actual. Esta estructura se denomina *backtrace (rastreo hacia atrás)*.[9]
Si no se utiliza ningún modelo de lenguaje, el mejor predecesor de una palabra para todas las palabras que empiezan en un *frame* es la palabra con la mejor puntuación en su estado de final de

palabra para el *frame* anterior. En ese caso, el *rastreo hacia atrás* tiene el mismo aspecto para todas las palabras y podría ser reemplazado por un simple indicador por *frame*. Si, pese a ello, el mejor predecesor depende de la identidad de la palabra debido a un modelo de lenguaje de *dos letras (bigram)*, hay que almacenar un *indicador retrospectivo (backpointer)* por cada *frame* y por cada final de palabra.
La información que debe almacenarse en el rastreo hacia atrás hasta alcanzar el último estado de una palabra podría obtenerse siguiendo los punteros situados en esa palabra de estado a estado. Sin embargo, para el *frame* en curso resulta mucho más eficiente mantener la información de la palabra de entrada al *frame* y la palabra que la precede en la estructura de la información de cada estado. En el siguiente *frame*, cada estado en el seno de la palabra hereda esta información de su mejor estado predecesor. Así, no es necesario almacenar más información sobre la ruta exacta entre las palabras. Los requisitos de memoria se reducen a la empleada para el rastreo hacia atrás, más los contenidos de todos los estados del *frame* en curso y el previo.
Las hipótesis de frases incompletas que están en construcción en cada uno de los estados de la cadena Markov se llaman *hipótesis parciales*. Muchas hipótesis parciales disponen de puntuaciones acumulativas que son tan bajas comparadas con otras hipótesis que no es probable que sean la mejor opción al final de la frase. Estas hipótesis parciales pueden descartarse del espacio de búsqueda.

Primera búsqueda profunda
Todos los algoritmos mencionados hasta ahora son síncronos en el tiempo y, por tanto, se destinan básicamente a búsquedas generales iniciales. La primera búsqueda profunda, sin embargo, guarda todas las hipótesis parciales en un fajo y expande sólo las que parecen más prometedoras en ese momento (Paul, 1992b; Paul y Neioglu, 1993). Una vez que la hipótesis alcanza el final del *frame* de la expresión, quedan disponibles las mejores hipótesis generales. El algoritmo puede ponerse en marcha bien para detenerse acto seguido o para proseguir hasta que la segunda mejor hipótesis alcance el último *frame* y, más adelante, hasta que la lista *N-best* resultante es lo suficientemente larga para la aplicación en cuestión.
El principal problema con la primera búsqueda profunda es saber qué hipótesis es la más prometedora, porque las puntuaciones en curso para las distintas hipótesis parciales pueden basarse en un número de *frames* variable. Asimismo, la organización eficiente de las hipótesis puede ser muy exigente. En consecuencia, las primeras búsquedas profundas no suelen ser el método elegido para el primer paso de un decodificador. Algunas veces son vistas como un segundo paso o se utilizan para combinar los resultados de varios pasos de búsqueda con nuevas fuentes de información.

Cuestiones de velocidad en el reconocimiento

Modelos de lenguaje
Aunque el proceso de computación de las probabilidades no es una cuestión intrínseca de velocidad, emplear modelos de lenguaje complica considerablemente el proceso de reconocimiento. Tres aproximaciones permiten reducir el impacto de los modelos de lenguaje en el tiempo de reconocimiento.

• *Búsqueda.* Encontrar la mejor secuencia de palabras dentro de un gran espacio de búsqueda es una tarea exigente en términos de computación. Los métodos como la purga de ese espacio y el uso de un léxico organizado en árbol son habituales para reducir el esfuerzo. A menudo, las mejoras en la estructura de búsqueda reducen también el tiempo requerido para otras subtareas.

• *Observación de probabilidades.* La mayoría de los sistemas LVCST invierten entre el 60 y el 90 % del tiempo de CPU en el cálculo de probabilidades de observación con mezcla de gaussianos, dependiendo del tamaño del vocabulario y de la complejidad de los modelos acústicos empleados. Esta sección presenta algunos algoritmos más rápidos para calcular estas probabilidades de observación y una aproximación alternativa por medio de redes neuronales.

• *Computación paralela y hardware.* Si las aproximaciones algorítmicas no resultan suficientes,

cabe recurrir a máquinas más rápidas o a hardware especial para estos fines.

Conclusiones

Un gran número de compañías están entrando en el mercado del habla de grandes vocabularios con considerable esfuerzo. Para que un software pueda comercializarse la velocidad en el reconocimiento del habla es tan importante como la precisión. Recientemente, algunas empresas han desarrollado sistemas LVCSR que aportan una gran fiabilidad con un rendimiento cercano al tiempo real. Con todo, estos sistemas difieren de forma significativa de los sistemas JRTk.
Los sistemas de reconocimiento rápidos basados en JRTk derivan directamente de sistemas de evaluación completa. Así, los tiempos de reconocimiento reducidos ayudan también a reducir el ciclo de tiempo para el desarrollo de sistemas de investigación, lo que debería llevar a que los reconocedores de voz ofrezcan precisiones aún superiores y mayores velocidades.
A largo plazo, algunos de los algoritmos presentados en esta tesis pueden quedar obsoletos. Por ejemplo, no está claro si las mezclas de gaussianos y redes neuronales son la mejor aproximación para estimar las probabilidades de observación. Con todo, la impresión actual es que esta combinación puede dar paso a las más bajas tasas de errores, resultado de una continua demanda de algoritmos como el *generalizado-BBI*.
A menos que un gran descubrimiento en los patrones de correspondencia introduzca un paradigma completamente nuevo para el problema de la búsqueda, el árbol de búsqueda y los algoritmos mencionados seguirán siendo útiles a largo plazo.
Finalmente, emplear una vista en modo dinámico del flujo de datos permitir efectuar cambios en la tasa de *frame*. Incluso puede ser un paso importante hacia sistemas más rápidos y más precisos en la selección de características dentro de una misma expresión.

Notas

Los ejemplos expuestos hacen referencia y se encuentran en la tesis de Monika Woszczyna:

1, 2, 3, 4 http://www.is.cs.cmu.edu/~monika/thesis/html/node11.htm

5 http://www.is.cs.cmu.edu/~monika/thesis/html/node12.htm

6 http://www.is.cs.cmu.edu/~monika/thesis/html/node13.htm

7 http://www.is.cs.cmu.edu/~monika/thesis/html/node15.htm

8 http://www.is.cs.cmu.edu/~monika/thesis/html/node20.htm

9 http://www.is.cs.cmu.edu/~monika/thesis/html/node22.htm

Monika Woszczyna

Trabaja desde 1991 en el Interactive Systems Laboratories, en la School of Computer Science de la Carnegie Mellon University en Pittsburgh y la University of Karlruhe en Alemania, donde ha realizado estudios de reconocimiento y traducción del habla con el profesor Alex Waibel. Durante largo tiempo ha estado trabajando en el Departamento de Ciencia computacional de la Karlsruhe University, alternando esta actividad con estancias regulares en la Carnegie Mellon University de Pittsburgh.
monika@cs.cmu.edu