UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DEPARTAMENTO DE INGENIERIA ELECTRICA

CONVERSION DE TEXTO A VOZ MEDIANTE REGLAS Y REDES NEURONALES: TRADUCCION DE TEXTO A FONEMAS MAS ACENTUACION Y PUNTUACION

MEMORIA PARA OPTAR AL TITULO DE INGENIERO CIVIL ELECTRICISTA

ROBERTO IGNACIO SMITH TORRES

PROFESOR GUIA
CLAUDIO PEREZ FLORES

MIEMBROS DE LA COMISION:
PABLO ESTEVEZ VALENCIA
CLAUDIO HELD BARRANDEGUY

SANTIAGO DE CHILE
ABRIL 2009

UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FISICAS Y MATEMATICAS
DEPARTAMENTO DE INGENIERIA ELECTRICA

CONVERSION DE TEXTO A VOZ MEDIANTE REGLAS Y REDES NEURONALES: TRADUCCION DE TEXTO A FONEMAS MAS ACENTUACION Y PUNTUACION

MEMORIA PARA OPTAR AL TITULO DE INGENIERO CIVIL ELECTRICISTA

ROBERTO IGNACIO SMITH TORRES

PROFESOR GUIA
CLAUDIO PEREZ FLORES

MIEMBROS DE LA COMISION:
PABLO ESTEVEZ VALENCIA
CLAUDIO HELD BARRANDEGUY

SANTIAGO DE CHILE
ABRIL 2009

RESUMEN DE LA MEMORIA PARA
OPTAR AL TÍTULO DE INGENIERO
CIVIL ELECTRICISTA
POR: ROBERTO SMITH T.
FECHA: 27/04/2009
PROF. GUÍA: DR. CLAUDIO PÉREZ F.

CONVERSIÓN DE TEXTO A VOZ MEDIANTE REGLAS Y REDES NEURONALES: TRADUCCIÓN DE TEXTO A FONEMAS MÁS ACENTUACIÓN Y PUNTUACIÓN

Para facilitar el acceso de las personas no-videntes al contenido de un texto se han desarrollado diversos sistemas, tanto mecánicos como electrónicos. De todos ellos, los lectores computarizados de textos han demostrado presentar mayores ventajas en cuanto a su facilidad de uso, cobertura, costo y calidad. Además, permiten el acceso directo a textos de publicación periódica, como diarios o páginas web. Los de mejor calidad de síntesis descomponen el problema de generación de voz en etapas sucesivas para resolver problemas como: acentuación, conversión de texto a fonemas, puntuación, incorporación de entonación, y síntesis de voz. En particular, para las etapas de acentuación y conversión de texto a fonemas, se han utilizado varias metodologías como: redes neuronales, reglas por defecto, pronunciación por analogía y análisis morfológico. En esta memoria se desarrollaron dos métodos alternativos para la conversión de texto a fonemas: redes neuronales y reglas por defecto.

Existe un sistema llamado NETtalk en el que se desarrolló un método de conversión de texto a fonemas para el idioma inglés, que utiliza redes neuronales de tipo perceptrón de múltiples capas. En este trabajo de título se realizó una adaptación de NETtalk para conversión de texto a fonemas más acentuación en el español hablado en Chile. Se usaron tres arquitecturas de red: traducción a fonemas, traducción más acentuación, y sólo acentuación. Se elaboraron conjuntos para entrenamiento, validación y prueba, utilizando para ello criterios basados en reglas fonéticas y ortográficas. El desempeño de la red que traduce texto a fonemas y acentúa, medido en el conjunto de prueba, fue 100% en traducción y 95,8% en acentuación. Se detectaron dos causas de error: conjunto de entrenamiento poco diverso y segmentos de palabras que se escriben igual pero tienen diferente acentuación. Para mejorar el desempeño de las redes se propone incorporar al conjunto de entrenamiento palabras con casos no contemplados y separar la traducción y la acentuación en redes especializadas. Se detectó empíricamente que las últimas 6 letras definen la vocal acentuada de cualquier palabra. Para la acentuación se propone usar una red que reciba esas 6 letras y codifique la posición del acento en la capa de salida. Para la traducción a fonemas, se propone usar una red que reciba 3 letras, que es la información necesaria para detectar el fonema mediante reglas.

Alternativamente se desarrolló un método de conversión de texto a fonemas del español chileno más acentuación utilizando reglas fonéticas y ortográficas. El desempeño obtenido fue 100% en los tres conjuntos de palabras, mejor que lo obtenido con redes neuronales.

Se midió el tiempo requerido para traducir el conjunto de entrenamiento de 1491 palabras a fonemas más acentuación mediante reglas, en un computador Pentium II de 350MHz con 288 MB de RAM, y fue 0,11 segundos, mientras que la red neuronal demora 620 veces ese tiempo.

Se desarrolló una aplicación en C como apoyo a la lectura de no-videntes, controlada íntegramente a través del teclado, llamada Asistente de Lectura, que convierte texto a fonemas con acentuación mediante reglas y redes neuronales. Para realizar síntesis de voz se utilizó un paquete de desarrollo de software para Windows 3.1 de Creative Labs. La aplicación permite abrir y traducir archivos de texto a fonemas, controlar la reproducción del texto y las características de la voz sintetizada. Al evaluar la calidad de síntesis con 10 usuarios se concluyó que la voz de español chileno presenta alta inteligibilidad pero es deficiente en naturalidad. Por ello se recomienda crear una nueva aplicación, utilizando las herramientas y estándares gratuitos para el diseño de conversores de texto a voz, y realizar un estudio detallado de las características fonéticas y prosódicas del español chileno para elaborar una base de datos de voz de buena calidad.

AGRADECIMIENTOS

Agradezco al proyecto FONDECYT 1960921, a mi profesor guía Claudio Pérez por su gran paciencia y por tener siempre una disposición abierta para que llevara este trabajo hasta el final, a los profesores de la comisión Pablo Estévez y Claudio Held, por su sentido del humor, por el tiempo y energía dedicados y la buena disposición. Agradezco muy especialmente a mis padres, Marta y Carlos, por su apoyo, su amor y sus hermosos sueños, a mis hermanos Carlos, Pablo y especialmente a mi hermana Marta quien me dio el espacio para dar término a la parte gruesa de este documento. Agradezco a la larga lista de personas que me apoyaron directa o indirectamente en este trabajo: Feli, Vilma, Vanel, Héctor Véliz, Jorge Atala, María Elisa Bazán, Pablo Alarcón, Alejandro Alonso, a las funcionarias que han estado presentes en las distintas etapas de mi proceso de titulación, María Elena y Eliana, a los compañeros de viaje: Erasmo, Millarca, Claudiana, Lucho Hamm, Tamara, a Poly por su gran cariño y apoyo, a la Ame y Chanchurro y la familia Davagnino, a la comunidad franco-chilena: Eric, Guillaume, Olivier, Sylvie, Karliña, Gato, Ina; al Taller del Adulto Menor, especialmente a Andriu, Karina, Daniel, Pame, Clau y Juanca, y a todas las personas que estuvieron vinculadas a esta larga historia y que por olvido no mencioné. Agradecimientos especiales a Pame, Mario y Auca por mostrarme el camino del ser, y recordarme que se puede todo lo que uno quiere desde el corazón.

ÍNDICE

1. INTRODUCCIÓN

1.1 Introducción

Con el objeto de elaborar mecanismos de apoyo a la lectura para no videntes, se han desarrollado diversos sistemas, tanto mecánicos como electrónicos: Braille (libros, impresoras), cintas magnéticas de audio, softwares (reglas, diccionarios de excepciones, redes neuronales). Cada uno de estos sistemas tiene algunos aspectos positivos y otros negativos respecto de su versatilidad y su accesibilidad para el usuario. El sistema Braille no es accesible a todos los no-videntes ya que requiere de un aprendizaje previo. Contrariamente a lo que se cree, sólo un 10% de los no-videntes es capaz de utilizar este tipo de lenguaje [1]. El alto costo y la escasa disponibilidad de textos en Braille hacen más inaccesible su uso. Por último, desde un punto de vista práctico, cabe señalar que un diccionario como "El Pequeño Larousse" traducido al Braille requiere de 157 volúmenes que ocupan 15 m lineales en estantería [1].

Otra alternativa de apoyo para la lectura de no-videntes es el acceso a cassettes de audio pregrabados. Hay instituciones (en Chile, el Centro de Grabación para Ciegos) donde personas sin problemas de vista se dedican a grabar textos en cintas de audio. La limitación de este método es el reducido número de libros disponibles.

Sin embargo, ninguna de estas dos alternativas permite a un no-vidente acceder en forma directa a un texto y, en particular, a textos de publicación periódica, como diarios o revistas. Es por esto que la creación de nuevos sistemas lectores computarizados de textos convencionales constituye una necesidad indispensable: su accesibilidad, menor costo y amplia cobertura representan ventajas indiscutibles.

En este ámbito se han desarrollado diversas aplicaciones, algunas de las cuales utilizan la síntesis de voz. Muchas de ellas presentan falencias en la calidad de la síntesis, en la rapidez y/o en la facilidad de su operación, o en el alto costo de los equipos que incorporan [1, 2].

Este trabajo se enmarcó dentro de un proyecto de investigación iniciado en 1994 en que se desarrolló un sistema que permite adquirir caracteres de texto a través de una cámara de video. Este consta de un módulo de reconocimiento que utiliza redes neuronales y que entrega como resultado los caracteres de texto reconocidos [3, 4, 5].

Se requiere adicionar la modalidad de utilizar síntesis de voz para computadores personales, empleando conversores análogo digitales [6]. Por otra parte, se han utilizado redes neuronales en la generación de voz para representar asociaciones de intensidad variable entre entrada y salida [7]. Existen sistemas híbridos para controlar la generación de voz, que utilizan redes neuronales de tipo perceptrón de múltiples capas y representación de conocimiento basado en reglas por defecto [8]. En otros trabajos se han desarrollado sintetizadores de voz basados en circuitos generadores de fonemas[1] ingleses, modificados para reconstruir los españoles [9, 10]. A modo experimental se han desarrollado módulos de síntesis de voz en español sobre la base de fonemas en inglés, usando redes neuronales del tipo mencionado; este sistema no incorpora fonemas en español, acentuación ni puntuación.

[1] Fonema: la forma de onda de la voz humana puede ser representada en forma discreta como una serie de segmentos, llamados fonemas. Cada idioma tiene un número determinado de fonemas a partir de los cuales se puede generar la forma de onda de cualquier palabra [11].

En otros trabajos se ha propuesto descomponer el problema de generación de voz en etapas sucesivas que permiten resolver problemas tales como: acentuación, conversión de letras a fonemas, incorporación de entonación (prosodia), puntuación, y sintetización de sonido.

Los desarrollos mencionados procuran obtener un lector de textos para no-videntes.

1.2 Antecedentes

Dentro de los desarrollos de lectores computarizados de textos, se destacan los siguientes:

TEXTTalk [11]: este es un sistema de conversión de texto a voz elaborado en Inglaterra por la empresa British-Telecom para el idioma inglés. Para realizar la síntesis de voz, el texto original pasa por varios procesos previos que generan un conjunto de parámetros para controlar el hardware de producción de sonido. Estos procesos son cuatro:

- Módulo de restricción de texto: consiste en un diccionario para identificar abreviaciones y acrónimos comunes y una lista de combinaciones posibles de consonantes. Las combinaciones no permitidas por el idioma son pronunciadas letra por letra.
- Módulo de pronunciación: consiste en un diccionario de excepciones, un detector de prefijos y sufijos, un módulo de traducción de texto a fonemas mediante reglas, y un módulo de acentuación mediante reglas.
- Módulo de prosodia: selecciona el alófono [2] más adecuado para cada fonema, y le asigna duración y tono de acuerdo a reglas basadas en el contexto que rodea al fonema.
- Módulo de generación de parámetros: traduce el resultado de los módulos anteriores a comandos y parámetros que el hardware de sonido pueda interpretar.

En resumen, el sistema consta de un gran conjunto de reglas y diccionarios con listas de excepciones.

DECtalk [12]: es un producto comercial que produce síntesis de voz inteligible en un dominio restringido para el idioma inglés. Usa dos métodos para convertir el texto a fonemas: primero las palabras son buscadas en un diccionario de pronunciación y, si no son encontradas, entonces se aplica un conjunto de reglas fonológicas. Los fonemas con acentuación resultantes son convertidos a voz mediante reglas de transición y síntesis digital de voz.

NETtalk [13]: este desarrollo, a diferencia de los anteriores, no utiliza reglas para la traducción del texto a fonemas ni para su acentuación. Esto es porque el idioma inglés tiene, para una misma letra, una gran cantidad de sonidos diferentes, dependiendo del contexto en que dicha letra se encuentre. Por ello, elaborar un sistema de reglas que entregue el sonido o fonema correcto para todas las combinaciones en que se presenta cada letra es muy complejo.

---

[2] Alófono: en la voz humana, un mismo fonema puede tener variaciones en su pronunciación, dependiendo de cuales son los fonemas que lo rodean, de la posición de la palabra a la que pertenece el fonema dentro de la frase y de los signos de puntuación. Estas distintas manifestaciones de un mismo fonema reciben el nombre de alófonos [11].

[3] Prosodia: consiste en asignar duración, frecuencia y volumen a cada fonema, con el objeto de generar la entonación adecuada de cada frase en base a los signos de puntuación que la delimitan [11].

NETtalk es un sistema basado en redes neuronales del tipo perceptrón de múltiples capas, que utiliza el algoritmo de aprendizaje de retropropagación del error. La red se usa para realizar la traducción de texto a fonemas y para la acentuación. El conocimiento en este modelo está distribuido sobre las unidades de procesamiento de la red, y el comportamiento de ésta frente a una entrada específica es una decisión colectiva basada en el intercambio de información entre las unidades de procesamiento.

TextAssist [14, 15, 16]: es un conjunto de aplicaciones comerciales de síntesis de voz desarrolladas para Windows por CREATIVE LABS para sus modelos de tarjeta de sonido Sound Blaster 16 ASP y posteriores. De todas estas aplicaciones, la más relevante relacionada con la síntesis de voz es Texto'LE [14, 15, 16], que está basada en el sistema DECtalk [12]. Esta permite realizar síntesis de voz a partir de un texto. Para ello usa un diccionario interno, un diccionario de excepciones, y reglas de pronunciación de lenguaje para convertir cada segmento de texto en fonemas con información de duración y frecuencia. El siguiente módulo convierte los fonemas en parámetros de control, que son utilizados para producir la síntesis de voz mediante el hardware. El texto puede ser escrito directamente en una ventana de edición de la misma aplicación, o puede ser obtenido a partir de un archivo. Los procesos realizados sobre el texto para generar la voz que lo pronuncia son transparentes al usuario, es decir, no hay acceso al conjunto de fonemas, acentuación, comandos de pausa y de prosodia previos a la pronunciación. TextAssist incorpora un paquete de desarrollo no comercial llamado TAAPI, que consiste en una biblioteca de funciones para desarrollar aplicaciones de síntesis de voz en Windows 3.1. Por el hecho de ser funciones independientes, entrega la flexibilidad de diseño de acuerdo a las necesidades del programador.

En el trabajo de título que se describirá en los siguientes capítulos fueron de gran importancia los dos últimos desarrollos descritos: NETtalk, que fue tomado como base para el método de traducción de texto a fonemas y de acentuación mediante redes neuronales; y TAAPI de TextAssist, que fue utilizado para la síntesis de voz una vez que el texto ya está traducido a fonemas con acentuación. En la aplicación elaborada como lector de texto para no-videntes, se integraron en un mismo programa el módulo de conversión de texto a fonemas más acentuación de TextAssist y los dos métodos desarrollados en este trabajo, uno basado en reglas y otro en redes neuronales. A diferencia de NETtalk y TextAssist, la aplicación desarrollada en esta memoria de título es para el idioma español chileno en vez del inglés. Está orientada a usuarios no-videntes por lo que todos los comandos de control se acceden a través del teclado. Los conjuntos de entrenamiento, validación y prueba utilizados para entrenar las redes neuronales y para evaluar el desempeño de las redes y las reglas fueron desarrollados en este trabajo de título, ya que las bases de datos del sistema NETtalk eran para el idioma inglés.

1.3 Objetivos

El objetivo general de este trabajo de título es desarrollar un sistema de conversión de texto a voz que permita obtener, a partir de un texto en español, una salida audible inteligible en el mismo idioma.

Los objetivos específicos consideran desarrollar un módulo de conversión de texto a fonemas en español chileno que incorpore acentuación y puntuación, empleando para ello tanto reglas ortográficas como redes neuronales. Por otra parte, utilizando un software de síntesis de voz y el módulo de conversión de texto a fonemas se propone desarrollar una aplicación computarizada que opere como lector de texto con una interfaz apropiada para usuarios no-videntes.

1.4 Contexto de este trabajo

Este trabajo fue comenzado el año 1996, como continuación de un desarrollo previo realizado por Héctor Véliz, alumno de Ingeniería Civil Electricista de la Universidad de Chile, enmarcado en el proyecto de investigación FONDECYT 1960921, en el que se estaba desarrollando un sistema que permitía adquirir caracteres de texto a través de una cámara de video. Este constaba de un módulo de reconocimiento que utiliza redes neuronales y entregaba como resultado los caracteres de texto reconocidos [3, 4, 5]. El traspaso de información de ese proyecto se realizó mediante una comunicación personal durante los años 1996 y 1997.

El objetivo de este trabajo de título fue desarrollar un lector computarizado de texto en español chileno para apoyo a la lectura de no-videntes. Para ello se disponía del software TextAssist [14], de Creative Labs, que realiza síntesis de voz en español e inglés en el sistema operativo Windows 3.1 para plataformas de hardware PC compatibles a través de una tarjeta de sonido SBAWE32 Creative. Este sistema de síntesis se eligió por ser el más accesible en términos de costo, requerimientos de hardware (espacio en disco duro y memoria RAM), y por estar diseñado para Windows 3.1, que era el sistema operativo más utilizado en el momento de la elección. Para convertir texto a fonemas, el software TextAssist recibe dos tipos de entrada: archivos de texto ASCII y archivos con texto fonetizado, este último compuesto por comandos que corresponden a los fonemas que se van a reproducir además de la ubicación de los acentos y de los signos de puntuación.

TextAssist presenta un problema: al entregar un archivo de texto ASCII, la salida de voz en español utiliza fonemas que no se usan en Chile, lo que produce un sonido poco natural e inteligible. Por otra parte, los fonemas en español incluidos en TextAssist tienen un sonido de menor calidad que los del inglés. Para solucionar este problema, fue necesario convertir el archivo de texto ASCII en un archivo de texto fonetizado con los fonemas adecuados, es decir, convertir previamente el archivo de texto ASCII a fonemas del español chileno. Para la conversión de texto a fonemas en inglés existía un sistema llamado NETtalk basado en redes neuronales del tipo perceptrón de múltiples capas. En este trabajo de título se continuó un desarrollo previo de implementación de este sistema para el español chileno. Se implementó, además, en forma paralela, un sistema de conversión de texto a voz basado en reglas ortográficas.

Se desarrolló para los sistemas operativos Windows 3.1, Windows 95 y Windows 98, un lector computarizado de texto en español chileno llamado Asistente de Lectura, para lo cual se utilizó el paquete de desarrollo de TextAssist. El Asistente de Lectura tiene tres alternativas para la conversión de texto a voz: reglas, redes neuronales, y conversión mediante las funciones que el mismo paquete de desarrollo provee. El sistema puede utilizar fonemas en español o sus equivalentes en inglés, éstos últimos con un sonido más nítido pero con acento inglés.

Tanto los antecedentes y objetivos así como el desarrollo y las conclusiones descritas en este documento están enmarcados en el contexto de avance científico y tecnológico de la fecha en que se realizó el trabajo.

Adicionalmente a lo descrito anteriormente, se hicieron los siguientes desarrollos adicionales entre los años 2006 y 2007 para efectos de actualizar y mejorar este trabajo de título producto del tiempo transcurrido:

- Conjuntos de palabras
    - Se creó un nuevo conjunto de palabras (conjunto de prueba) para obtener una medida del desempeño de las redes que fuera independiente del proceso de entrenamiento

    - Se modificaron los conjuntos de entrenamiento y validación, incorporando palabras nuevas con el objeto de equilibrar, dentro de las restricciones del lenguaje, los distintos casos que debían estar representados en ellos, tanto para la acentuación como para la traducción a fonemas
- Modificación de las arquitecturas de redes: En el trabajo original se utilizó una arquitectura de red para cada objetivo (traducción a fonemas, acentuación y traducción más acentuación). Como mejora al desarrollo se buscó disminuir el tamaño de las redes sin empeorar el desempeño de éstas, para disminuir el tiempo requerido en obtener las salidas de las redes. Para ello se realizaron los siguientes cambios:
    - Número de capas ocultas: se eliminó una de las dos capas ocultas, disminuyendo el tiempo de procesamiento
    - Se realizaron pruebas variando para cada objetivo (traducción a fonemas, acentuación y traducción más acentuación) el número de unidades de la capa oculta (80, 30, 20, 10, 7 y 6) de modo de obtener el tamaño mínimo manteniendo el desempeño
- Se realizaron 6 entrenamientos por cada arquitectura de red para medir la incidencia de las condiciones iniciales en el desempeño de las redes

1.5 Estructura de la Memoria de Título

Esta memoria está organizada en cuatro capítulos y una serie de anexos. El primer capítulo es introductorio; el contenido de los siguientes se resume a continuación:

En el capítulo 2 se detallan los dos métodos utilizados en este trabajo para convertir texto en fonemas: reglas y redes neuronales. Para el método de reglas, se enumeran las características fonéticas del español y las particularidades de su habla en Chile y se describen las reglas que se deducen para la conversión de texto a fonemas. Se detallan, además, las reglas ortográficas que permiten detectar cuál es la vocal que tiene acento en las palabras acentuadas implícitamente (sin acento gráfico).
Para el método de redes neuronales, se describe la estructura de las redes neuronales, las modificaciones que se realizan sobre la red NETtalk, que fue tomada como base para este desarrollo, y los criterios considerados para elaborar los conjuntos de entrenamiento, validación y prueba para cada una de las redes. Además, se explica, a grandes rasgos la manera en que se programaron tanto las reglas como las redes neuronales.
Se describe el Asistente de Lectura, que es la aplicación que se desarrolló como apoyo a la lectura de no-videntes. En la primera parte, se detallan las funcionalidades del programa y las distintas opciones y comandos para realizarlas. En la segunda parte, se describe la forma en que se realizó el programa: plataforma de desarrollo, lenguaje de programación y subrutinas más importantes.
Finalmente, se describen las pruebas realizadas para medir el desempeño de los métodos de conversión de texto a fonemas y del Asistente de Lectura.

En el capítulo 3 se muestran los resultados obtenidos en las pruebas y se hace un análisis de ellos.

En el capítulo 4 se presentan las conclusiones del trabajo, en donde se resumen los resultados más relevantes obtenidos para la conversión de texto a fonemas, los errores y sus posibles causas. Se proponen mejoras a los métodos implementados y se hace una recomendación, basada en los desarrollos actuales más difundidos y con mejor calidad de síntesis, en cuanto a las decisiones de metodología, hardware y software para el desarrollo de un nuevo sistema computarizado de conversión de texto a voz en español chileno como interfaz de apoyo a la lectura de textos por parte de usuarios no-videntes.

Se incluyen cuatro anexos. El primero es un resumen sobre los avances de los últimos años en los sistemas de conversión de texto a voz y en las aplicaciones de hardware y software de apoyo a la lectura de no videntes. El segundo anexo es un resumen sobre redes neuronales y el algoritmo de retropropagación utilizados en las redes implementadas en este trabajo. El tercer anexo es una descripción del paquete de aplicaciones computacionales TextAssist, que se utilizó para generar síntesis de voz. El cuarto y último anexo es una lista con las palabras de los conjuntos de entrenamiento, validación y prueba de las redes neuronales.

2. METODOLOGÍA

En este capítulo se describe la metodología utilizada para cumplir con los objetivos específicos de este trabajo de título: implementación de dos mecanismos alternativos para la traducción de texto a fonemas (reglas y redes neuronales) y diseño y desarrollo de una aplicación de síntesis de voz como apoyo a la lectura de no videntes utilizando los dos mecanismos de traducción de texto a fonemas mencionados.

2.1 Síntesis de voz y traducción de texto a fonemas

Para desarrollar lectores computarizados que lleven a cabo síntesis de voz a partir de un texto, es decir, para simular una voz humana que lo lee, se requieren varias etapas: traducir el texto a fonemas, incorporar la acentuación de las palabras, convertir las puntuaciones a pausas, y controlar la prosodia. Esta última consiste en asignar duración, frecuencia y volumen a cada fonema, con el objeto de generar la entonación adecuada de cada frase en base a los signos de puntuación que la delimitan. Además, se requiere de software y hardware adecuado para transformar el conjunto de información de fonemas, acentuación, puntuación y prosodia en sonido.

Existen varias metodologías para realizar cada etapa. Dos de ellas son utilizadas en este trabajo de título para la conversión de texto a fonemas: redes neuronales y reglas.

Las redes neuronales se han aplicado con éxito a una variedad de áreas relacionadas con el reconocimiento y clasificación de patrones. Es por esto que es de interés investigar su aplicación en la síntesis de voz, particularmente en la traducción de texto a fonemas y en la acentuación. Los lenguajes que poseen alfabeto presentan las siguientes características que los hacen candidatos para utilizar redes neuronales en esas etapas de la síntesis: un conjunto finito de símbolos del alfabeto y un conjunto finito de fonemas o sonidos.

En la traducción de texto a fonemas, el problema consiste en decidir, para un símbolo del alfabeto en un contexto particular (una letra de una palabra), cuál es el fonema asociado; en la acentuación, el problema se trata de decidir, para ese mismo símbolo, si lleva acento o no. Ambos problemas corresponden a clasificación de patrones y es por ello que se pueden abordar en base a un sistema de reglas así como con redes neuronales.

Con respecto a las redes neuronales, existe un desarrollo basado en ellas para el idioma inglés, llamado NETtalk [13], que es capaz de traducir texto a fonemas con acentuación y puntuación en base a la clasificación de patrones. Este desarrollo se describe más adelante en este capítulo y fue tomado como base para realizar un sistema similar en el idioma español.

Alternativamente, los idiomas con alfabeto permiten utilizar conjuntos de reglas y excepciones para la traducción de texto a fonemas y para la acentuación. Cada idioma tiene diferentes reglas que rigen ambos procesos y la eficiencia en su desempeño depende de la regularidad con que dichas reglas se cumplan, es decir, el número de palabras que obedezcan estas normas dentro del total de vocablos del lenguaje correspondiente. El español presenta características más favorables que el inglés tanto para la traducción del texto a fonemas, como para la acentuación, debido a que los sonidos asociados a las palabras son menos irregulares. Esto permite realizar ambas etapas del procesamiento mediante un conjunto de reglas.

A continuación se describen los métodos de traducción de texto a fonemas y acentuación mediante el uso de reglas. Posteriormente, se explica la manera de abordar los mismos problemas utilizando redes neuronales.

2.1.1 Traducción de texto a fonemas y acentuación mediante reglas

Traducción de texto a fonemas

Una de las etapas previas a la generación de voz a partir de un texto escrito, es la traducción del texto a los fonemas que tiene asociados.

En el alfabeto español hay 27 símbolos o letras diferentes:

a b c d e f g h i j k l m n ñ o p q r s t u v w x y z

La mayoría de estos símbolos tienen sólo un fonema asociado. Sin embargo, para un grupo de ellos, el fonema depende de los símbolos que están en torno suyo, presentándose casos en que incluso no tienen fonema asociado, es decir, no generan sonido. Estas situaciones pueden cubrirse con un conjunto de reglas ortográficas [7].

Por otra parte, algunos símbolos tienen asociado un fonema diferente, dependiendo del país, aunque se trate siempre del habla hispana. En Chile, hay letras que, siendo distintas, en su pronunciación se escuchan idénticas, mientras que en otros países se distinguen notoriamente entre sí.

Considerando tanto las reglas ortográficas mencionadas como las características de pronunciación del español en Chile, se realizan modificaciones al conjunto de símbolos del abecedario, con el objeto de eliminar las ambigüedades en los sonidos asociados a cada uno. De este modo, para traducir un texto a fonemas, basta con aplicar las modificaciones sobre dicho texto, es decir, traducirlo al abecedario modificado.

La Tabla 1 resume la lista de las modificaciones del abecedario, y su justificación. En la columna izquierda se muestran las letras que pueden presentar distintos sonidos dependiendo del contexto que las rodea. La segunda columna describe el contexto para el cual se presenta cada posible sonido asociado a la letra. En la tercera columna se indica el sonido asociado a la letra y, en la cuarta columna, el símbolo que se utiliza en este trabajo para representar dicho sonido, es decir, el abecedario modificado. Finalmente, la última columna muestra un ejemplo que representa el caso descrito.

En resumen, se eliminan siete símbolos del abecedario: “c, h, q, v, w, x, z.”, y se agregan dos: “ç”, que corresponde al sonido “ch”, y “R”, que corresponde al sonido “rr”. Luego, el abecedario modificado queda con los siguientes 22 símbolos:

a b ç d e f g i j k l m n ñ o p r R s t u y

Tabla 1: Letras del abecedario para el español de Chile con sonidos modificados o no obvios. En la columna LETRA se indican las letras que requieren aclaración. En la columna CASO se describen los casos para los cuales se presenta cada posible sonido asociado a la letra. En la columna SONIDO se indica el sonido (en términos alfabéticos) asociado a la letra para cada caso y, en la columna SÍMBOLO se muestra el símbolo que se utiliza en este trabajo para representar dicho sonido. Finalmente, en la última columna se incluye una palabra de ejemplo para representar el caso descrito.

LETRA	CASO	SONIDO	SÍMBOLO ASOCIADO	EJEMPLO
c	antes de h	ch	ç	Choque
	antes de e y de i	s	s	Cecilia
	después de x **(1)**	-	-	Excepción
	cualquier otro caso	k	k	crocante
g	antes de e o de i	j	j	género
	cualquier otro caso	g	G	gato
h	cualquier caso	-	-	
l	antes de l	y	y	calle
	después de l	-	-	calle
	cualquier otro caso	l	l	cola
q	cualquier caso	k	k	dique
r	cuando es 1ra letra de la palabra o cuando va después de l, n, r ,s	rr	R	enroque
	antes de r	-	-	carro
	cualquier otro caso	r	r	arnés
u	después de g o q y antes de e o i	-	-	quequito, guinda
	cualquier otro caso	u	u	cuento
ü	cualquier caso	u	u	ambigüedad
v	cualquier caso	b**(2)**	b	vaca
w	cualquier caso	u**(3)**	u	wanda
x	cualquier caso	ks	k+s	exceso
y	cuando es última letra de palabra	i	i	buey
	en cualquier otro caso	y	y	rayo
z	cualquier caso	s**(4)**	s	cazar

Notas:

1: un caso particular es la letra “c” cuando va precedida de “x”. Como se ve en el recuadro, la “x” corresponde a la concatenación de otras dos letras: “k” y “s”, por lo que puede ser eliminada como símbolo. Por ello, en casos como la palabra *excelente*, el resultado al traducir sería *eksselente*. Para evitar la doble “s”, se elimina la “c” precedida de una “x”.

2: en español chileno, la “v” se pronuncia idéntico que la “b”. Además, el software TextAssist tiene un mismo fonema para la “b” y la ”v”.

3: en español, tiene sonido de “u”.

4: en español chileno se pronuncia idéntico que la “s”. Sin embargo, como se verá más adelante, este alfabeto modificado se utiliza también para procesar el texto antes de ser entregado a una red neuronal, que se encarga de acentuar las palabras. Las letras “s” y “z” definen si la posición del acento implícito de una palabra va en la última o penúltima sílaba, por lo que, para efectos de la red, no se elimina del abecedario modificado, aunque al ser traducida a fonemas, se traduce como “s”.

Acentuación

Para realizar síntesis de voz, además de la traducción de texto a fonemas, es necesario saber dónde se acentúan las palabras, pues influye en la manera en que se pronuncian.

Descripción del problema

Desde el punto de vista de la acentuación, las palabras del español se pueden clasificar por el tipo de acento y por la sílaba acentuada.

a) Tipo de acento: todas las palabras tienen acento. Este se ubica siempre en una sola vocal, y es en ella donde se carga la voz al pronunciar la palabra. De acuerdo a su representación gráfica, los acentos se clasifican como [18]:

(def1) Explícito: se representa por una raya diagonal, inclinada hacia la derecha, y se ubica sobre la vocal acentuada.
(def2) Implícito: no tiene representación gráfica.

b) Sílaba acentuada: una sílaba se define como "sonido o sonidos articulados que constituyen un solo núcleo fónico entre dos depresiones sucesivas de la emisión de voz" [18]. Una sílaba tiene como mínimo una vocal y como máximo tres; en caso de ser dos, se dice que forman diptongo; en caso de ser tres, triptongo. Las palabras se pueden separar en cuatro grupos, de acuerdo a la sílaba en la cual se encuentra la vocal acentuada:

(def3) Agudas: llevan el acento en la última sílaba. Un caso particular de las agudas son las palabras con una sílaba o monosílabos
(def4) Graves: llevan el acento en la penúltima sílaba.
(def5) Esdrújulas: llevan el acento en la antepenúltima sílaba.
(def6) Sobre esdrújulas: llevan el acento en la sílaba anterior a la antepenúltima.

Las palabras esdrújulas y sobre esdrújulas, por regla ortográfica, se acentúan explícitamente. Las graves y agudas se acentúan en forma explícita o implícita de acuerdo a lo que indiquen las reglas ortográficas y sus excepciones.

Por lo tanto, el problema se reduce a la identificación de la vocal acentuada en las palabras con acento implícito, es decir, a los casos de graves y agudas sin acento gráfico.

Solución del problema

Para abordar el problema se utilizan dos reglas generales y dos excepciones que rigen la acentuación de las palabras graves y agudas, de acuerdo a la Gramática de la Lengua Española [18]:

(1) Agudas: llevan acento explícito cuando terminan en letra "n", "s" o vocal.
(2) Graves: llevan acento explícito cuando terminan en consonante distinta de "n" y "s".
(3) Existe una excepción a la regla (1), para los monosílabos, es decir, las palabras que tienen una sola sílaba. Estas palabras son, por definición, agudas, pues van acentuadas en la última sílaba. Sin embargo, de acuerdo a las reglas ortográficas de la gramática española, no tienen acento explícito, salvo algunos pronombres personales y otras excepciones, aún cuando terminan en letra "n", "s" o vocal.
(4) Existe una segunda excepción: la presencia de hiatos (descritos más adelante, en la definición 10) puede provocar acentuación explícita en agudas terminadas en

consonantes distintas de “n” o “s”, y en graves terminadas en “n”, “s” o vocal. Sin embargo, ya que se trata de casos en que el acento es explícito, no hay incertidumbre respecto de su acentuación.

De acuerdo a estas reglas, sólo llevan acento implícito las siguientes palabras:

(1a) Agudas terminadas en letra consonante distinta de “n” y de “s”, sin hiato.
(2a) Graves terminadas en vocal o terminadas en alguna de las consonantes “n” o “s”, sin hiato.
(3a) Monosílabos que no son pronombres personales ni pertenecen al conjunto de excepciones de la regla (3).

Por lo tanto, para acentuar las palabras de un texto, el problema se reduce a la detección de las vocales acentuadas implícitamente para el grupo de palabras anteriormente descrito.

Para saber cuál es la vocal que tiene acento en una palabra acentuada implícitamente, son necesarios tres pasos:

a) Identificación del tipo de palabra: grave, aguda, y/o monosílabo.
b) Identificación del número de vocales de la sílaba acentuada.
c) Identificación de la vocal acentuada.

a) Identificación del tipo de palabra: grave, aguda, y/o monosílabo

Para detectar a qué tipo corresponde la palabra acentuada implícitamente, se verifica primero si es aguda no monosílaba; si no lo es, se verifica si es monosílabo. En caso de no ser aguda ni monosílabo, se trata de una palabra grave.

- Agudas no monosílabos: de acuerdo a las reglas (1), (2) y (3), son todas las palabras que terminan en consonante distinta de “n” o “s”

Para las palabras que no cumplen esta condición, es decir, las que terminan en “n”, “s“ o vocal, hay que detectar si se tratan de graves o monosílabos, es decir, hay que conocer el número de sílabas de la palabra. Para ello, se utilizan las siguientes reglas ortográficas:

(5) Una sílaba tiene una vocal como mínimo, y tres como máximo.
(6) Si la sílaba tiene más de una vocal, deben estar juntas.

Por lo tanto, en las palabras en que las vocales están rodeadas por consonantes, no hay dificultad para determinar a qué sílaba pertenece cada una. Donde existe duda es en aquellas palabras en que el acento se encuentra en una vocal que está unida a una o más vocales en torno suyo.

Para describir y analizar por separado dichas palabras, son necesarias las siguientes definiciones o clasificaciones de las vocales:

(def7) Vocal fuerte: son las vocales “a”, “e” y “o”.
(def8) Vocal débil: son las vocales “i” y “u”.

Además, en base a esta clasificación de las vocales, se definen las siguientes combinaciones:

(def9) Diptongo: cuando una vocal es débil y la otra es fuerte, o ambas débiles, y se presentan en la misma sílaba.

(def10) Hiato: cuando una vocal es débil y la otra es fuerte, o ambas débiles, y se presentan en sílabas diferentes.
(def11) Triptongos: cuando se produce alguna de las siguientes combinaciones de vocales débil-fuerte-débil dentro de una misma sílaba: iai, iei, uai, uei.

Por lo tanto, en las palabras acentuadas implícitamente y que presentan dos o más vocales juntas, se puede producir diptongo, hiato, dos vocales fuertes y triptongos.

Cuando la vocal acentuada pertenece a alguno de estos casos, existen reglas de acentuación que permiten deducir cuál de las vocales involucradas es la que tiene el acento:

(7) Si la sílaba acentuada tiene un diptongo, el acento va en la vocal fuerte. Si ambas son débiles, va en la segunda vocal.
(8) Si la vocal acentuada pertenece a un hiato, el acento es explícito y va en la vocal débil; por lo tanto, no pertenece al conjunto de palabras en estudio.
(9) Si las dos vocales son fuertes, ambas van en sílabas separadas.
(10) Si la sílaba acentuada tiene un triptongo, el acento va en la vocal fuerte.

En base a todas estas definiciones y reglas, se puede identificar a qué corresponden las palabras sin acento explícito que no son agudas:

- Monosílabos: son monosílabos todas las palabras que tienen sólo una vocal. Son monosílabos, también, en virtud de las reglas (5), (6) y la definición 9, las palabras con sólo dos vocales que, además, forman diptongo, es decir, ambas están juntas y una es débil y la otra fuerte, o las dos débiles. Finalmente, de acuerdo a la definición 11, son monosílabos las palabras con sólo tres vocales siempre que formen triptongo, es decir, cuando las tres están juntas en orden débil-fuerte-débil.

- Graves: Son graves todas las palabras que no son agudas y/o monosílabos.

b) Identificación del número de vocales de la sílaba acentuada

Para identificar si la sílaba acentuada tiene más de una vocal, se analizan por separado las palabras monosílabos, agudas y graves.

- Monosílabos: es el caso más sencillo, ya que el número de vocales de la sílaba acentuada es el número de vocales de la palabra.

- Agudas: basta con analizar la última sílaba. El número de vocales de la sílaba acentuada es 1 en los siguientes casos: si la última vocal está rodeada de consonantes, de acuerdo a la regla (5), o si es una vocal fuerte rodeada de una consonante y otra vocal fuerte, de acuerdo a la regla (9). El número de vocales es 2 si las últimas dos vocales forman diptongo, por lo que tienen que cumplir con la definición 9. Por último, el número de vocales es 3 si las tres últimas vocales forman triptongo, en cuyo caso deben cumplir con la definición 11.

- Graves: En el caso de las palabras graves, que por definición se acentúan en la penúltima sílaba, puede suceder que las vocales de la sílaba acentuada estén junto con las vocales de la última sílaba. Si es así, hay que determinar el límite entre las dos últimas sílabas antes de poder identificar el número de vocales de la penúltima sílaba. Para ello, se recorre la palabra desde el final hacia el comienzo. La primera vocal que aparece pertenece, por la regla (5), a la última sílaba. Se producen dos casos: la vocal es débil o es fuerte.

La vocal es débil: dependiendo de cual sea la siguiente letra que aparezca siguiendo el recorrido de la palabra, se producen tres situaciones:

- Consonante: en este caso, de acuerdo a las reglas (5) y (6), la siguiente vocal que aparezca pertenece a la penúltima sílaba, y a continuación se puede utilizar el mismo procedimiento que en las agudas para determinar el número de vocales de la penúltima sílaba.

- Vocal débil: en este caso se trata de un diptongo, de acuerdo a la definición 9. A continuación, se puede utilizar el mismo procedimiento que en las agudas para determinar el número de vocales de la penúltima sílaba, recorriendo ahora desde la letra anterior a esta vocal débil.

- Vocal fuerte: dependiendo de cual sea la siguiente letra que aparezca siguiendo el recorrido de la palabra, se producen tres situaciones:

  - Consonante: en este caso, las dos vocales encontradas forman un diptongo, de acuerdo a la definición 9. En virtud de las reglas (5) y (6), la siguiente vocal que aparezca pertenece a la penúltima sílaba, y a continuación se puede utilizar el mismo procedimiento que en las agudas para determinar el número de vocales de la penúltima sílaba.

  - Vocal fuerte: en este caso, las dos primeras vocales encontradas forman un diptongo, de acuerdo a la definición 9. La nueva vocal pertenece a la penúltima sílaba, de acuerdo a la regla (9). A continuación, se puede utilizar el mismo procedimiento que en las agudas para determinar el número de vocales de la penúltima sílaba, recorriendo ahora desde la última vocal fuerte encontrada.

  - Vocal débil: en este caso, las tres vocales encontradas forman un triptongo, de acuerdo a la definición 11. A continuación, se puede utilizar el mismo procedimiento que en las agudas para determinar el número de vocales de la penúltima sílaba, recorriendo ahora desde la letra anterior a la vocal débil encontrada.

La vocal es fuerte: dependiendo de cual sea la siguiente letra que aparezca siguiendo el recorrido de la palabra, se producen tres situaciones:

- Consonante: en este caso, de acuerdo a las reglas (5) y (6), la siguiente vocal que aparezca pertenece a la penúltima sílaba, y a continuación se puede utilizar el mismo procedimiento que en las agudas para determinar el número de vocales de la penúltima sílaba.

- Vocal fuerte: en este caso ambas pertenecen a sílabas diferentes, de acuerdo a la regla (9). No existen casos de tres vocales fuertes juntas ni de una débil seguida de dos fuertes. Por lo tanto, el número de vocales de la segunda sílaba es uno.

- Vocal débil: en este caso, sin importar cual sea la vocal final, se trata de un diptongo, de acuerdo a la definición 9. A continuación, se puede utilizar el mismo procedimiento que en las agudas para determinar el número de vocales de la penúltima sílaba.

c) Identificación de la vocal acentuada

Una vez que se ha determinado cuál es la sílaba acentuada y el número de vocales que contiene, hay que identificar cuál es la vocal acentuada. Existen tres casos:

- Sílabas con una vocal: la vocal acentuada en la única vocal de la sílaba.

- Sílabas con dos vocales: hay dos combinaciones, dependiendo del tipo de vocal.

Dos vocales débiles: se acentúa la segunda vocal, de acuerdo a la regla (7).

Una vocal débil y una fuerte: se acentúa la vocal fuerte, de acuerdo a la regla (7).

- Sílabas con tres vocales: tienen la forma débil - fuerte - débil, y el acento va en la vocal fuerte, de acuerdo a la regla (9).

En la siguiente sección se muestra el pseudocódigo para la acentuación mediante reglas.

Pseudocódigo para la acentuación mediante reglas

- Si la palabra está acentuada explícitamente (Ej.: tú, ésta, aún)
    - Fin
- Si palabra no está acentuada explícitamente (Ej.: dos, voy, fue, marea, guagua, callen)
    - Si la palabra tiene una sola vocal (Ej.: dos)
        - Si la palabra termina en “y” (Ej.: voy)
            - Se acentúa la vocal
        - Fin (Ej.: dos, vóy)
    - Si la palabra tiene más de una vocal
        - Si la palabra es grave o monosílabo (No acentuada y terminada en n, s o vocal) (Ej.: fu*e*, calle*n*, diario*s*, mare*a*)
            - Si la letra anterior a la última vocal es una vocal fuerte (mar*e*a)
                - Si la letra anterior a esa vocal fuerte no es una vocal débil
                    - Esa vocal fuerte se acentúa (maréa)
                    - Fin
                - Si la letra anterior a esa vocal fuerte es una vocal débil
                    - Si la palabra no tiene más vocales
                        - Se acentúa la vocal fuerte.
                        - Fin
                    - Si la palabra tiene sólo una vocal más
                        - Esa vocal va acentuada
                        - Fin
                    - Si la palabra tiene por lo menos dos vocales más
                        - Si la última de esas dos vocales es fuerte
                            - Se acentúa esa vocal
                            - Fin
                        - Si la letra anterior a esa vocal es una vocal fuerte
                            - Esa vocal fuerte va acentuada
                            - Fin
                        - Si la letra anterior a esa vocal es una vocal débil
                            - La última de esas dos vocales va acentuada
                            - Fin
            - Si la letra anterior a la última vocal es una vocal débil o consonante(f*u*e, lai*c*o)
                - Si la palabra no tiene más vocales
                    - Se acentúa última vocal (fué)
                    - Fin
                - Si la palabra tiene sólo una vocal más (c*a*llen, *a*gua)
                    - Esa vocal va acentuada (cállen, água)
                    - Fin
                - Si la palabra tiene por lo menos dos vocales más (guagua, faena, laico)
                    - Si la última de esas dos vocales es fuerte (gu*a*gua, fa*e*na)
                        - Se acentúa esa vocal (guágua, faéna)
                        - Fin
                    - Si la letra anterior a esa vocal es una vocal fuerte (l*a*ico)
                        - Esa vocal fuerte va acentuada (láico)
                        - Fin
                    - Si la letra anterior a esa vocal es una vocal débil (ag*ü*ita)
                        - La última de esas dos vocales va acentuada (agüíta)
                        - Fin
        - Si la palabra es aguda y no monosílabo (No acentuada, terminada en consonante distinta de n o s) (Ej.: pasto*r*, Ariel, real)
            - Se acentúa la última vocal (pastór, Ariél, reál)
            - Fin

Implementación

Reglas para la traducción a fonemas con acentuación

Para implementar las reglas de traducción de texto a fonemas y de acentuación, se utilizaron como ambientes de desarrollo DOS y Windows 3.1, y se usó el compilador de Borland C (Turbo C 1.0 y Turbo C/C++ 3.1). La plataforma de hardware fue un computador PC compatible con procesador Pentium II de 350MHz, con 288 MB de memoria RAM y sistema operativo Windows 98.

Primero se realiza el proceso de acentuación de acuerdo al pseudocódigo descrito. Para ello, cada palabra es recorrida desde la última letra hacia el comienzo, hasta detectar cuál es la vocal acentuada.

Una vez acentuado el texto, se procede a la traducción a fonemas, siguiendo las reglas resumidas en la Tabla 1. Para traducir una letra al fonema adecuado, basta con tomar en cuenta los caracteres que la rodean y, en los casos que sea necesario, aplicar la regla correspondiente.

Pruebas

Para medir el desempeño de las reglas en la acentuación y la traducción a fonemas, se utilizó la aplicación Asistente de Lectura desarrollada para este trabajo de título, en la que se incorporó la funcionalidad de traducción y acentuación a fonemas utilizando los algoritmos de reglas descritos. Se realizó la siguiente prueba: acentuación y traducción a fonemas de los conjuntos de entrenamiento, validación y prueba, de 1.491, 135 y 135 palabras, respectivamente. Se midió el porcentaje de aciertos y el tiempo empleado en el proceso. Se midió el número de caracteres de cada conjunto antes y después de ser traducido a fonemas. La estructura, composición y criterios de construcción de los tres conjuntos de palabras se describen en la subsección “Conjuntos de entrenamiento, validación y prueba” de la sección 2.1.2.

2.1.2 Traducción de texto a fonemas y acentuación mediante una red neuronal

En la sección anterior se revisó la conversión de texto a fonemas y la acentuación por medio de reglas. En esta sección se describe cómo se realizaron ambos procesos utilizando redes neuronales. Como se mencionó en la introducción de este capítulo, los lenguajes que poseen alfabeto tienen un conjunto finito de símbolos y un conjunto finito de fonemas o sonidos, características que los hacen candidatos para utilizar redes neuronales en esas etapas de la síntesis. Las tareas de la red son:

Acentuación: clasificar cada letra del texto de entrada como acentuado o no acentuado.
Traducción a fonemas: clasificarla dentro de alguno de los fonemas del español de Chile.

El método desarrollado en este trabajo para traducir texto a fonemas mediante redes neuronales está basado en el sistema NETtalk [13], que emplea redes neuronales del tipo perceptrón de múltiples capas y utiliza el algoritmo de aprendizaje de retropropagación.

Estructura de la red neuronal del sistema NETtalk

En esta sección se describe el sistema NETtalk [13] para el idioma inglés. Para realizar traducción y acentuación se utiliza una red neuronal de tres capas: una capa de entrada, una capa oculta y una de salida, como se muestra en la figura 1.

En la definición de la estructura de la red se consideran aspectos relacionados con la acentuación y con la traducción a fonemas.

Acentuación: todas las palabras del inglés tienen acento implícito en una vocal. Si la palabra tiene más de una vocal, se requiere conocer las letras que la rodean para saber en cuál está el acento.

Traducción a fonemas: algunas letras del abecedario tienen un sonido u otro dependiendo de cuales son las letras que las rodean. Por ello, para clasificar correctamente, la red debe tener como entrada la palabra completa o, por lo menos, una cantidad suficiente de letras en torno a la que es estudiada.

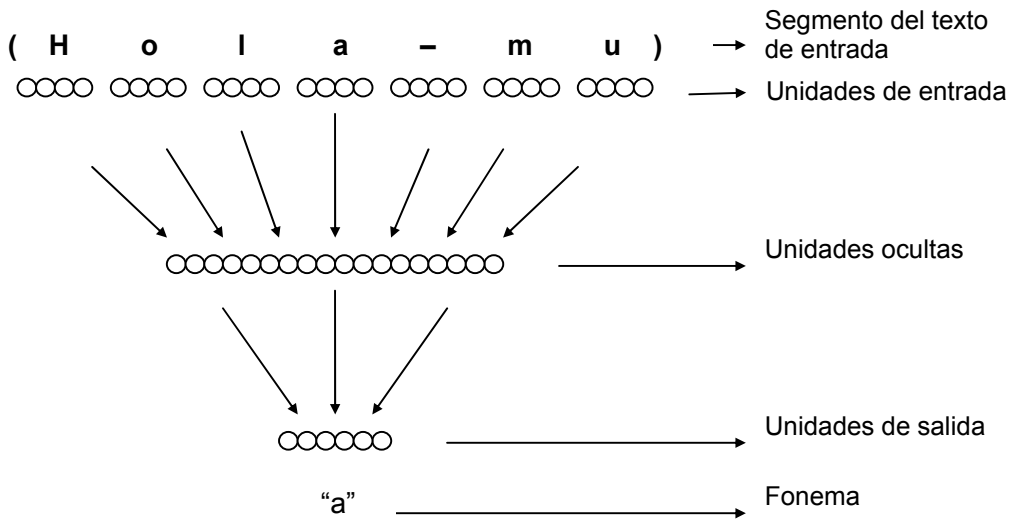


FIGURA 1: Red neuronal tipo perceptrón multicapa utilizado en NETtalk. La entrada corresponde a una ventana de 7 letras de texto. La salida codifica el fonema asociado a la letra central de la ventana de entrada.

En las redes del tipo perceptrón de múltiples capas el número de entradas es fija. Este es el caso de NETtalk. Como las palabras tienen un largo variable, para definir el número de entradas de la red sería necesario considerar la palabra más larga posible. Para evitar este sobredimensionamiento, se considera una ventana de entrada de la siguiente manera:

1 2 ... N * 1 2 ... M

donde la letra que se va a clasificar (traducir y/o acentuar) está representada por un *. Las entradas de la red son las N letras anteriores a * y las M letras siguientes. En el sistema

NETtalk se considera una ventana de siete letras, con N = M = 3. Esta ventana formada por 7 letras consecutivas se desplaza desde la primera hasta la última letra del texto, clasificando cada una como acentuada o no acentuada y asignándole alguno de los fonemas de la capa de salida.

Capa de entrada: La capa de entrada está compuesta por siete grupos de unidades. Cada grupo codifica una letra del texto a traducir, de modo que se entregan como entrada a la red grupos de siete letras consecutivas a la vez. La codificación de las letras es binaria, considerando tantas unidades como símbolos tiene el abecedario, que para el caso del inglés son 26. Para cada grupo que codifica una letra, la unidad que la representa toma valor 1, mientras que el resto de las unidades toma valor cero. Como es necesario separar las palabras, se agrega el símbolo "-", que representa un espacio o frontera entre ellas y no tiene sonido asociado. Se usan otras dos unidades adicionales para codificar las puntuaciones. Esto da un total de 29 unidades por cada grupo. En la figura 2 se presenta la codificación resultante para la letra "s".

```
    abcdefghijklmnopqrstuvwxyz-.:        (Símbolos del abecedario inglés)
s   00000000000000000010000000000        (Codificación para la letra "s")
```

FIGURA 2: Ejemplo de codificación binaria de la letra "s" en el sistema NETtalk.

Capa oculta: hay una sola capa oculta con 80 unidades.

Capa de salida: la capa de salida tiene 23 unidades para codificar los 51 fonemas del inglés considerados por el sistema. Por lo tanto, algunas unidades de salida participan en la codificación de más de un fonema. Además, se consideran tres unidades adicionales para codificar la acentuación, puntuación y los bordes de palabras, dando un total de 26 unidades. La salida de la red es el fonema asociado a la letra central de la ventana de siete letras, es decir, a la cuarta. El resto de las letras de la ventana proveen el contexto parcial para la decisión.

Para el entrenamiento de la red, se utilizaron dos textos: un trozo continuo de lenguaje informal de un niño, y un conjunto de 20.012 palabras de un diccionario. Con el primer texto, el procedimiento para entrenar la red fue desplazar la ventana letra a letra a través de todo el texto, generando la salida correspondiente para cada letra. De esta manera, varias palabras (con símbolos de separación entre ellas) o fragmentos de palabras podrían estar a un mismo tiempo dentro de la ventana. Para el diccionario, las palabras fueron entregadas aleatoriamente, y desplazadas individualmente a través de la ventana. La aplicación y los algoritmos de la red neuronal fueron programados en lenguaje C.

Para medir el desempeño de la red, se utilizó un conjunto de validación compuesto por subconjunto de palabras del diccionario, correspondiente a las de uso más común en el inglés, ya que presentan características más irregulares, permitiendo conocer la capacidad de la red de absorber excepciones.

Estructura de las redes neuronales utilizadas en este trabajo

Una primera adaptación de NETtalk para el español hablado en Chile fue desarrollada por Héctor Véliz, alumno de la carrera de Ingeniería Civil Electricista de la Universidad de Chile. Esta versión realiza traducción de texto a fonemas con acentuación mediante una red neuronal análoga a la utilizada por NETtalk, en donde se modificaron el número de símbolos del abecedario, el número de capas ocultas, y el número de fonemas, pues éste último es diferente

para cada idioma. También se modificó la forma en que se codifican los fonemas en la capa de salida.

Con el objeto de estudiar por separado el desempeño de las redes neuronales en la acentuación y en la traducción de texto a fonemas, en este trabajo, basándose en la adaptación de NETTalk, se implementaron redes para tres objetivos: 1) traducción a fonemas más acentuación; 2) sólo traducción a fonemas; 3) sólo acentuación. Para cada uno de estos objetivos se realizaron entrenamientos variando el número de unidades de la capa oculta, para identificar el tamaño mínimo de las redes sin empeorar el desempeño.

A continuación se describe la estructura de las capas de estas redes.

Capa de entrada

El número de unidades de la capa de entrada depende del número de símbolos del abecedario. En el alfabeto español hay 27 símbolos o letras diferentes:

a b c d e f g h i j k l m n ñ o p q r s t u v w x y z

Además, existen dos tipos de acentos: gráfico (**'**) y cremilla (¨), que afectan a las vocales y a la letra "u", respectivamente. Por ello, se agregan los siguientes símbolos de entrada de red: **ü** y **'**. Cuando se presenta una vocal acentuada gráficamente, en la codificación de la entrada para esa letra se activa la unidad que representa a la vocal y la unidad que representa al acento.

Para que la red sea capaz de distinguir entre el fin de una palabra y el comienzo de otra, se utiliza como separador un caracter espacio. Dependiendo de la forma en que se realice el entrenamiento, las palabras pueden estar separadas por uno o más de estos caracteres. Por ello, se agrega otro símbolo más para representar dicha separación: "**-**".

Finalmente, el conjunto de símbolos de entrada queda compuesto por los siguientes 30 elementos:

a b c d e f g h i j k l m n ñ o p q r s t u ü v w x y z ' -

Al igual que en NETtalk, se considera una ventana con 7 símbolos de entrada: el central es el que va a ser traducido y acentuado (si corresponde), y los dos grupos de tres símbolos que lo rodean forman el contexto necesario para la decisión.

La codificación de las entradas es análoga a la de NETtalk. Como el número de símbolos es 30, las unidades de la capa de entrada son:

$$7 * 30 = 210$$

El mismo ejemplo de codificación de la letra "s" en NETtalk, queda ahora para el español chileno como se muestra en la figura 3.

```
   abcdefghijklmnñopqrstuüvwxyz'-      (Símbolos de entrada)
s  000000000000000000010000000000      (Codificación para letra s)
```

FIGURA 3: Ejemplo de codificación binaria de la letra "s" en español.

En la figura 4 se muestra la entrada de la red para la letra "b" de la palabra "había" rodeada de dos caracteres de espacio.

	Ventana de entrada (210 unidades) abcdefghijklmnñopqrstuüvwxyz'-
-	000000000000000000000000000001
h	0000000100000000000000000000000
a	1000000000000000000000000000000
b	0100000000000000000000000000000
í	0000000010000000000000000000010
a	1000000000000000000000000000000
-	000000000000000000000000000001

FIGURA 4: Ejemplo de la entrada de red para la letra "b" de la palabra " había".

Capas ocultas

Las redes tienen una sola capa oculta. Para cada tipo de red (traducción, acentuación y traducción más acentuación) se implementaron arquitecturas con distinto número de unidades en la capa oculta: 80, 30, 20, 10, 7 y 6 unidades, con el objeto de determinar el menor tamaño de la capa oculta sin que el desempeño de la red disminuya.

Capa de salida

El número de unidades de la capa de salida es distinto para las tres arquitecturas de red. A continuación se describe la estructura de la capa de salida de cada arquitectura.

a) Red para traducción a fonemas más acentuación

El número de unidades de la capa de salida depende del número de fonemas. En el español que se habla en Chile, como se indicó en la sección 2.1.1, son los siguientes:

a b ç d e f g h i j k l m n ñ o p r R s t u y ,

en que el símbolo "ç" representa el sonido de la combinación de letras "ch", el símbolo "R" representa el sonido de la combinación "rr" y el símbolo "g" representa el sonido de la letra "g" en la palabra "gato".

Debido a que hay casos de letras que en algún contexto no tienen sonido asociado se agrega un fonema que indica que no hay sonido para esa entrada: "**:**".

Al igual que en la capa de entrada, se utiliza una salida que indica si la entrada es una separación de palabra, usando el mismo símbolo para su representación. También se emplea una salida de acento, que indica si la entrada va o no acentuada. En consecuencia, la capa de salida tiene 26 unidades que toman valores reales entre 0 y 1, representando directamente la activación de un fonema asociado y la acentuación correspondiente:

a b ç d e f g i j k l m n ñ o p r R s t u x y : - ' .

Para la letra "x", que tiene asociada la cadena de fonemas "ks", se incluye una salida extra. De esta manera la red activa sólo una salida de fonemas en vez de activar las salidas de fonemas "k" y "s".

A modo de ejemplo, la figura 5 muestra la ventana de entrada para la letra “b” de la palabra “había” centrada en la ventana entre dos caracteres espacio. Los valores de la derecha corresponden a la salida esperada de la red para esa entrada.

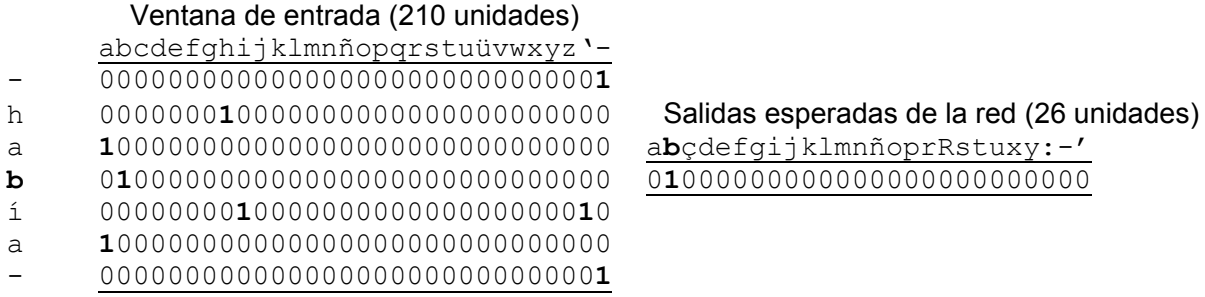


FIGURA 5: Ejemplo para la red que traduce y acentúa de la entrada y la salida esperada para la letra “b” de la palabra “había”.

b) Red para traducción a fonemas

La capa de salida es igual a la red que traduce y acentúa, pero se elimina la unidad de salida de acento. En la figura 6 se muestra cómo queda el ejemplo de la palabra “había” para esta red.

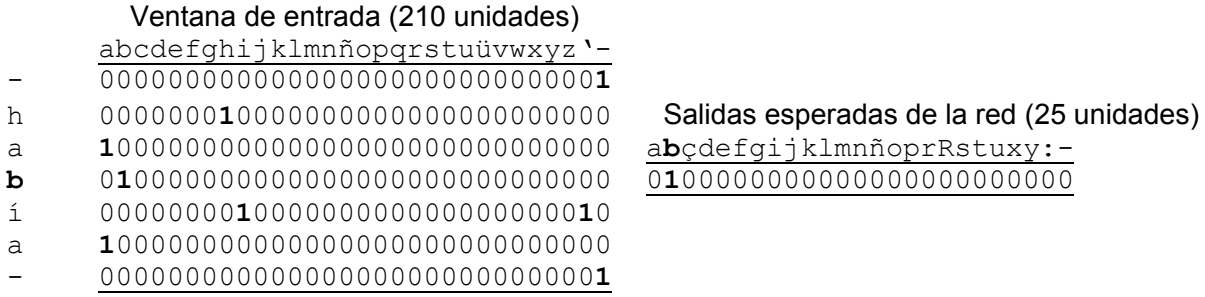


FIGURA 6: Ejemplo para la red que sólo traduce a fonemas de la entrada y la salida esperada para la letra “b” de la palabra “ había”.

c) Red para acentuación

La capa de salida tiene sólo la unidad de acento. En la figura 7 se muestra como queda el ejemplo de la palabra “había” para esta red.

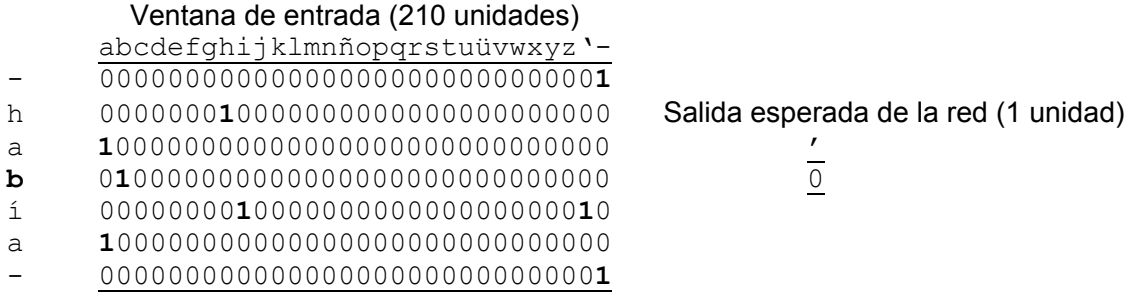


FIGURA 7: Ejemplo para la red que sólo acentúa de la entrada y la salida esperada para la letra “b” de la palabra “había”.

En resumen, la capa de entrada de las redes está compuesta por 210 unidades que se agrupan en 7 grupos de 30. Cada grupo se obtiene de la traducción de una letra a una palabra binaria de 30 dígitos. Estas palabras binarias representan las letras del alfabeto, considerando además un caracter para el espacio entre palabras y otro para el acento implícito, por lo que sólo toma valor 1 el dígito binario asociado a la letra y, si ésta va acentuada, también vale 1 el dígito asociado al caracter de acentuación. Hay una sola capa oculta, y se implementaron redes con 80, 30, 20, 10, 7 y 6 unidades. La conexión entre las unidades de una capa y la siguiente es uno a uno. Para la red de traducción y acentuación, la capa de salida tiene 26 unidades que toman valores reales entre 0 y 1. Cada unidad representa la activación, para una entrada, del fonema del español chileno asociado al caracter central de la ventana de 7 letras y su acentuación. Para la red que sólo traduce, se elimina la salida de acento por lo que la capa de salida tiene 25 unidades. Finalmente, la red que sólo acentúa tiene una sola unidad que representa la activación del acento.

En la las figuras 8, 9 y 10 se muestran ejemplos de esquemas de las arquitecturas de las redes neuronales para traducción de texto a fonemas más acentuación, sólo traducción a fonemas y sólo acentuación, respectivamente. El número de unidades de la capa oculta de cada ejemplo es 7, 30 y 30, respectivamente.

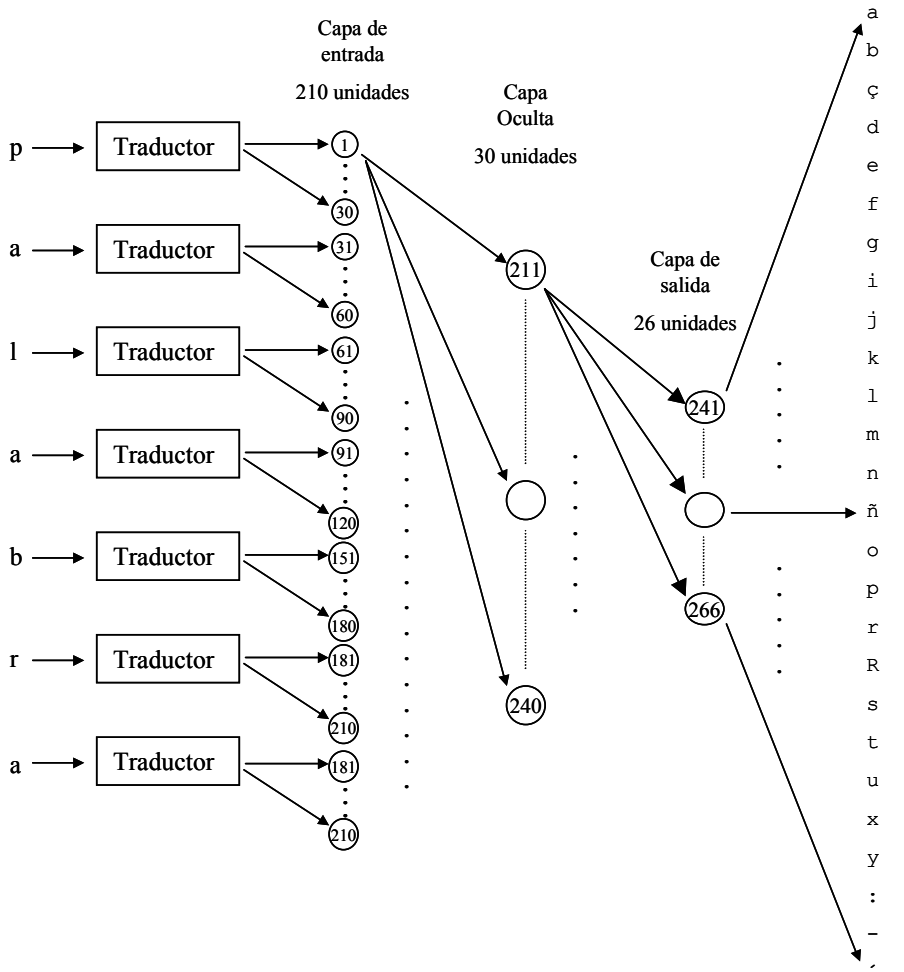


Figura 8: Arquitectura de la red neuronal que realiza acentuación y traducción de texto a fonemas.

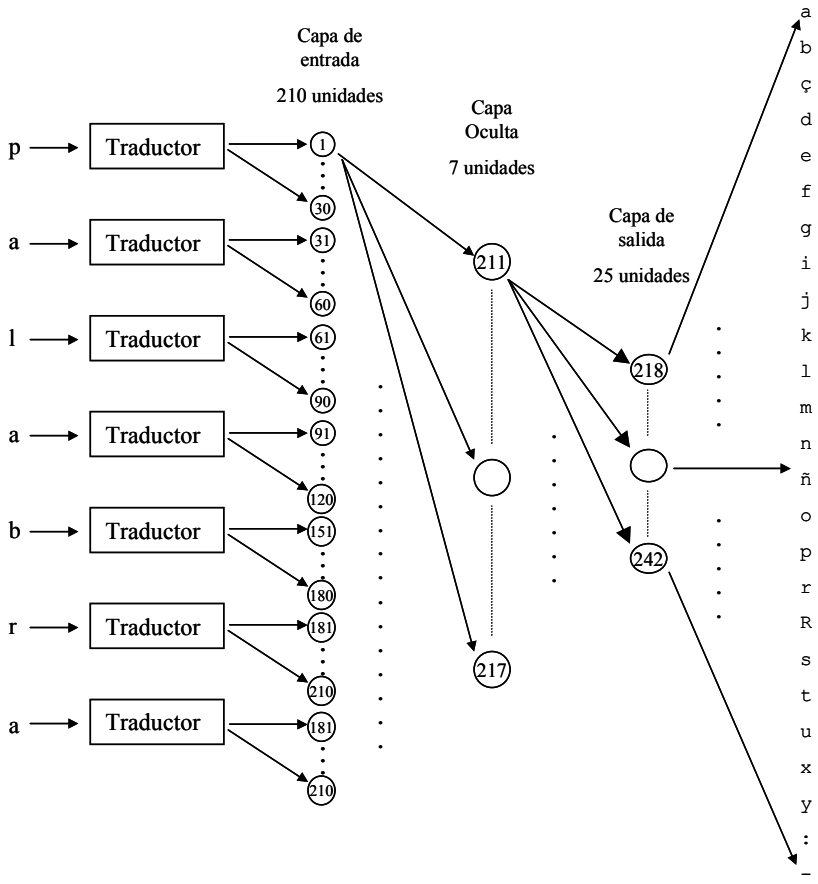


Figura 9: Arquitectura de la red neuronal que realiza sólo traducción de texto a fonemas.

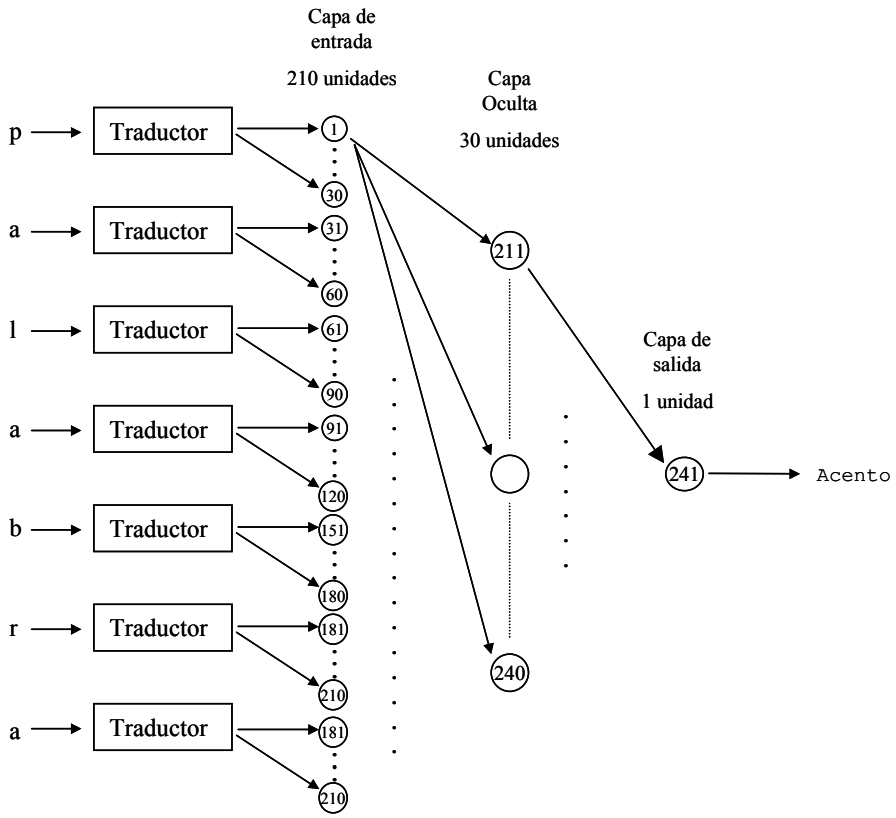


Figura 10: Arquitectura de la red neuronal que realiza sólo acentuación de texto.

Parámetros de aprendizaje

Los valores asignados a los parámetros de aprendizaje fueron tomados del sistema NETtalk y después reajustados en pruebas preliminares en un desarrollo previo realizado por H. Véliz. Después de ajustar estos parámetros, no se volvieron a modificar y centró la atención en las capas de entrada, oculta y de salida de las redes y en la construcción de los conjuntos de entrenamiento, validación y prueba. Los parámetros de aprendizaje fueron los mismos para todas las redes:

Moméntum:	0,3
Tasa de aprendizaje (LR):	0,2
Random Gate (cota para inicialización de valores aleatorios):	0,2
Máximo error absoluto (condición para retropropagación):	0,1

Estos parámetros son descritos brevemente en el Anexo B.

Conjuntos de entrenamiento, validación y prueba

En este trabajo de título se utilizaron tres conjuntos de palabras, que son iguales para cada red: entrenamiento, validación y prueba. El primero se utilizó para el entrenamiento de la red. El conjunto de validación se utilizó para medir el desempeño de la red después de cada época por el conjunto de entrenamiento, y para seleccionar la época para la cual se obtuvo mejor desempeño. Finalmente, el conjunto de prueba se utilizó para evaluar el desempeño de la red obtenida en la época seleccionada mediante el conjunto de validación. Esta medida de desempeño es independiente del proceso de entrenamiento ya que las palabras de este conjunto no han participado en el proceso de entrenamiento ni de selección de mejor época. El desempeño de las redes neuronales depende, en gran medida, de la manera en que se elaboran los conjuntos con datos que se utilizan para el entrenamiento. Por ello, es necesario considerar los dos objetivos de las redes implementadas en este trabajo al momento de diseñar los conjuntos: traducción de texto a fonemas y acentuación explícita de palabras con acento implícito. A continuación se presentan las consideraciones para estos dos fines.

Traducción de texto a fonemas

Se consideraron dos criterios al seleccionar palabras para el conjunto de entrenamiento en cuanto a la traducción de texto a fonemas: el número mínimo de veces que debe aparecer cada letra y el número mínimo de veces que deben aparecer los casos que presentan ambigüedad. Idealmente debería haber una distribución homogénea de estos casos pero, al utilizar palabras reales que deben cumplir simultáneamente con las distintas restricciones de ambos criterios y, además, con los criterios de acentuación que se describen más adelante, el número de veces que aparece cada símbolo en el conjunto de entrenamiento se ve afectado por el número de veces que se presenta cada símbolo en el vocabulario español (a modo de ejemplo, las vocales son los símbolos que más aparecen, ya que cada sílaba de una palabra en español tiene al menos una). Por esta razón, se incluyeron palabras inventadas que cumplen con las reglas ortográficas y fonéticas del español chileno (por ejemplo, weñed, jauvag), las cuales permitieron homogeneizar la presencia de los casos menos representados de cada uno de los criterios de construcción utilizados para la elaboración de los conjuntos de palabras, tanto para traducción de texto a fonemas como para acentuación.

a) Número de veces que aparece cada letra en el conjunto de entrenamiento

Al elaborar el conjunto, se intentó que cada símbolo del alfabeto debe apareciera una cantidad mínima de veces, de modo que la red lo reconozca cuando forme parte de la entrada. Como resultado de esto, en la Tabla 2 se muestra el número de veces que aparece cada símbolo del alfabeto en el conjunto de entrenamiento.

Tabla 2: Número de veces que aparece cada símbolo del alfabeto en la entrada del conjunto de entrenamiento.

Símbolo	N° veces	Símbolo	N° veces	Símbolo	N° veces	Símbolo	N° veces
a	976	**j**	107	**r**	797	**z**	109
b	122	**k**	87	**s**	375	**á**	102
c	565	**l**	537	**t**	325	**é**	123
d	308	**m**	227	**u**	556	**í**	171
e	848	**n**	520	**ü**	81	**ó**	132
f	100	**ñ**	107	**v**	91	**ú**	123
g	457	**o**	713	**w**	74		
h	137	**p**	142	**x**	103		
i	717	**q**	80	**y**	113		

b) Número de veces que aparece cada caso ambiguo en el conjunto de entrenamiento

Para saber cual es el sonido asociado a un símbolo de entrada se necesita tomar en cuenta el contexto, es decir, las letras que lo rodean. Por ello, se trató de incluir un número mínimo de veces los casos en que existe ambigüedad respecto del sonido asociado a alguna letra. Estos casos están detallados en la Tabla 1 en la sección 2.1.1. Como resultado de esto, en la Tabla 3 se indica el número de veces que se presenta cada caso en el conjunto de entrenamiento.

Tabla 3: Número de veces que se presenta cada caso ambiguo en el conjunto de entrenamiento.

"c"	N°	"g"	N°	"L"	N°	"r"	N°	"u"	N°	"y"	N°
ce[1]	45	**g**e[4]	32	l	395	**l**r[7]	29	**g**u[11,13]	33	y[15]	58
ci[1]	73	**g**i[4]	35	**l**l[6]	71	**n**r[7]	29	**g**ue[12]	35	y[16]	55
ca[2]	63	**g**a[5]	32			**r**r[7,8]	76	**g**ui[12]	36		
co[2]	99	**g**o[5]	34			**s**r[7]	27	**q**ue[12]	39		
cu[2]	36	**g**l[5]	32			r[7,9]	55	**q**ui[12]	41		
cc[2]	26	**g**n[5]	40			r[10]	505	u[14]	237		
cd[2]	29	**g**r[5]	34								
cl[2]	33	**g**u[5]	33								
cn[2]	32	**g**ue[5]	35								
cr[2]	34	**g**ui[5]	36								
ct[2]	32	**g**üe[5]	47								
ch[3]	63	**g**üi[5]	34								

Notas:

1: letra “c” con sonido de “s”.
2: letra “c” con sonido de “k”.
3: letra “c” con sonido de “ch”.
4: letra “g” con sonido de “j”.
5: letra “g” con sonido de “g” (Ej.: gato).
6: letra “l” con sonido de “ll” (Ej.: primera “l” de “lleno”).
letra “l” sin sonido (Ej.: segunda “l” de “lleno”).
7: letra “r” con sonido de “rr”.
8: letra “r” sin sonido. (Ej.: primera “r” de “carro”).
letra “r” con sonido de “rr”. (Ej.: segunda “r” de “carro”).
9: palabras que comienzan con “r” y que tienen, por ello, sonido de “rr”.
10: letras “r” con sonido de “r” (antecedidas de letras distintas de l, n, r y s, y seguidas de letra distinta de r).
11: letra “u” con sonido de “u”.
12: letra “u” sin sonido.
13: combinación “gu” seguida de letras distintas de “e” y de “i”, o sin letras a continuación.
14: letra “u” antecedida por letras distintas de “g” y de “q” , o sin ser antecedida.
15: letra “y” con sonido de “y”, es decir, antes del final de una palabra. (Ej.: ayer)
16: letra “y” con sonido de “i”, es decir, al final de una palabra. (Ej.: estoy)

Acentuación

A diferencia del inglés, el español no presenta tanta irregularidad en la acentuación, pudiéndose distinguir casos en los que se clasifican las distintas palabras, como se describió en la acentuación por reglas. Por esto no es adecuado elaborar el conjunto de entrenamiento

seleccionando las palabras más utilizadas en el español, como se hizo con el sistema NETtalk. Las palabras que componen el conjunto de entrenamiento deben cubrir idealmente en forma equilibrada los diferentes casos ortográficos que se presentan en la acentuación del español. Para ello, se consideraron los siguientes criterios: tipo de acento (explícito e implícito), sílaba acentuada, contexto, y casos especiales (monosílabos).

a) Tipo de acento: explícito e implícito

Como se describió en la acentuación por reglas, todas las palabras en el español tienen acento en una vocal. Este acento puede ser explícito o implícito. Sólo en el segundo caso es necesario identificar cual es la vocal acentuada, por lo que este tipo de palabras compone en forma mayoritaria el conjunto de entrenamiento. Sin embargo, ya que la red debe realizar además traducción de texto a fonemas, fue necesario incluir palabras con acento explícito, pues algunas tienen combinaciones de letras que no se encuentran en las palabras acentuadas implícitamente. La Tabla 4 muestra el número de veces que aparece cada tipo de acento en el conjunto de entrenamiento.

Tabla 4: Número de veces que se presenta cada tipo de acento en el conjunto de entrenamiento.

Acento explícito	N° veces	Acento implícito	N° veces
á	102	**a**	216
é	123	**e**	191
í	171	**i**	154
ó	132	**o**	144
ú	123	**u**	135

b) Sílaba acentuada

De acuerdo a las definiciones 3 a la 6 dadas la sección 2.1.1, las palabras en español se pueden separar en cuatro grupos según la sílaba en que están acentuadas: agudas, graves, esdrújulas y sobre esdrújulas. Por ello, incluyen palabras tratando que el conjunto de entrenamiento sea equilibrado respecto de los distintos tipos de palabras del idioma. Las palabras graves y agudas son las únicas que pueden tener acento implícito, es decir, presentan dificultad para determinar su acentuación. Es por ello que, adicionalmente, se deben cubrir los diferentes casos ortográficos de acentuación para este grupo de palabras. Estos casos están determinados por las reglas (1) y (2), y significa que en la selección de las palabras agudas y graves se debe considerar la letra en que termina la palabra. Se eligieron, por lo tanto, palabras tratando de cubrir todas las letras finales posibles en forma equilibrada (en promedio, 4 palabras para cada letra final), tanto para las agudas como para las graves.

Como se describe en las reglas (5) y (6), una sílaba tiene una vocal como mínimo, y tres como máximo. Si hay más de una, deben estar juntas. En las palabras en que las vocales están rodeadas por consonantes, no hay dificultad para determinar a qué sílaba pertenece cada una. Existen situaciones en que la separación entre una sílaba y otra es confusa, esto es, cuando hay dos o más vocales juntas en una misma palabra y, en especial, cuando el acento va en alguna de ellas. Es necesario incluir este tipo de casos, que corresponden a las palabras con diptongos, hiatos, pares de vocales fuertes y palabras que combinen alguno de los anteriores, descritos en las definiciones 7 a la 10 y en las reglas (7) a la (10). Se eligieron, por lo tanto, palabras tratando de cubrir en forma equilibrada los casos de diptongos, hiatos y pares de vocales fuertes, para que la red sea capaz de absorber estas irregularidades. Las Tablas 5, 6, 7

y 8 muestran el número de casos de palabras agudas acentuadas implícitamente, graves acentuadas implícitamente, agudas acentuadas explícitamente y graves acentuadas explícitamente, respectivamente. En cada tabla el número de casos se ordena por vocal acentuada y letra final de la palabra. Las celdas grises corresponden a hiatos, y las celdas con "-" son combinaciones para las que no existen palabras en el español.

Tabla 5: Número palabras de agudas acentuadas implícitamente en el conjunto de entrenamiento.

		Vocal acentuada				
		a	**e**	**i**	**o**	**u**
Letra final de la palabra	**d**	10	6	6	6	11
	g	3	3	3	3	3
	j	4	3	3	3	3
	l	18	5	6	5	5
	m	3	3	3	3	3
	r	30	16	10	7	5
	t	3	5	4	4	4
	x	3	3	3	3	3
	y	13	10	-	9	5
	z	5	5	5	5	6

Tabla 6: Número de palabras graves acentuadas implícitamente en el conjunto de entrenamiento.

		Vocal acentuada				
		a	**E**	**i**	**o**	**u**
Letra final de la palabra	**a**	29	31	28	16	28
	e	16	18	14	9	9
	i	6	4	6	5	5
	n	6	6	5	5	5
	o	31	38	30	28	18
	s	12	14	11	10	7
	u	4	3	3	4	3

Tabla 7: Número de palabras agudas acentuadas explícitamente en el conjunto de entrenamiento.

		Vocal acentuada				
		a	**e**	**i**	**o**	**u**
letra final de la palabra	**á**	10	-	-	-	-
	d	-	-	4	-	4
	é	-	30	-	-	-
	í	-	-	31	-	-
	l	-	-	4	-	4
	n	9	8	8	30	8
	ó	-	-	-	32	-
	r	-	-	4	-	5
	s	9	9	8	8	8
	ú	-	-	-	-	10
	z	0	0	4	0	3

Tabla 8: Número de palabras graves acentuadas explícitamente en el conjunto de entrenamiento.

		Vocal acentuada				
		a	**e**	**i**	**o**	**u**
letra final de la palabra	**a**	-	-	21	-	12
	d	5	6	5	5	5
	e	-	-	5	-	5
	l	6	5	5	5	6
	m	5	6	5	5	3
	n	-	-	5	-	5
	o	-	-	9	-	5
	r	7	7	6	4	6
	s	-	-	6	-	5
	x	4	4	4	4	4
	z	5	5	5	5	5

Notas:

1.- Celdas grises: hiatos (Agudas acentuadas explícitamente terminadas en consonante distinta de "n" y "s"; Graves acentuadas explícitamente terminadas en "n", "s" o vocal).
2.- Celdas con símbolo "-": no existen palabras del español con esa combinación. [18]

c) Número de veces que aparece cada letra en el conjunto de entrenamiento

Para saber si una letra lleva acento o no, se necesita tomar en cuenta el contexto, es decir, las letras que la rodean. Al elaborar el conjunto, se considera que cada símbolo del alfabeto debe aparecer una cantidad mínima de veces, idealmente en forma equilibrada, de modo que la red lo reconozca cuando aparezca.

d) Casos especiales

Los monosílabos son, por definición, palabras agudas. Sin embargo, de acuerdo a las reglas de la gramática española, no llevan acento explícito, salvo algunos pronombres personales y otras excepciones. Esta situación genera una contradicción en las reglas, ya que las agudas terminadas en “n”, “s” o vocal deben llevar acento explícito, pero al ser monosílabos no pueden tenerlo. Por lo tanto, se incluyeron además de las palabras agudas normales, palabras monosílabas, de modo de que la red fuera capaz de “aprender” estas excepciones. La Tabla 9 muestra el número de monosílabos en el conjunto de entrenamiento, desglosados por letra final.

Tabla 9: Número de monosílabos en el conjunto de entrenamiento terminados en la letra indicada.

Letra final	N° veces	Letra final	N° veces
a	2	**n**	18
b	2	**o**	3
e	4	**r**	11
f	2	**s**	10
i	5	**u**	3
j	2	**y**	15
l	5	**z**	9

Construidos con criterios similares, se obtienen los tres conjuntos: entrenamiento, validación y prueba. Las Tablas 10, 11 y 12 muestran cómo se desglosan las palabras del conjunto de entrenamiento, validación y prueba, respectivamente. Las palabras esdrújulas siempre llevan acento explícito, por lo que no aparece ningún valor en la celda correspondiente a acento implícito.

Tabla 10: Desglose de las palabras del conjunto de entrenamiento de acuerdo al tipo de acento y al tipo de sílaba acentuada.

	Monosílabos	Agudas	Graves	Esdrújulas	Total
Acento implícito	81	293	467	-	**841**
Acento explícito	10	250	230	160	**650**
Total	**91**	**543**	**697**	**160**	**1491**

Tabla 11: Desglose de las palabras del conjunto de validación de acuerdo al tipo de acento y al tipo de sílaba acentuada.

	Monosílabos	Agudas	Graves	Esdrújulas	Total
Acento implícito	4	21	79	-	**104**
Acento explícito	2	10	6	13	**31**
Total	**6**	**31**	**85**	**13**	**135**

Tabla 12: Desglose de las palabras del conjunto de prueba de acuerdo al tipo de acento y al tipo de sílaba acentuada.

	Monosílabos	Agudas	Graves	Esdrújulas	Total
Acento implícito	4	21	79	-	**104**
Acento explícito	2	10	6	13	**31**
Total	**6**	**31**	**85**	**13**	**135**

El número real de patrones de cada conjunto se obtiene al traducir los conjuntos de palabras al formato de entrada de las redes. La Tabla 13 muestra el número de palabras y de patrones para los conjuntos de entrenamiento, validación y prueba, que son los mismos para las tres redes.

Tabla 13: Número de palabras y de patrones de los conjuntos de entrenamiento, validación y prueba.

Conjunto	Número de palabras	Número de patrones
Entrenamiento	1491	10025
Validación	135	955
Prueba	135	1001

En el Anexo D se presentan los listados de palabras de los conjuntos de entrenamiento, validación y prueba utilizados.

Entrenamiento

Las redes neuronales utilizadas en este trabajo son del tipo perceptrón de múltiples capas y para su entrenamiento supervisado se emplea el método de retropropagación del error [17]. En este método se le entrega una entrada a la red, se obtiene la salida de la red para esa entrada, se compara con la salida esperada y, en base a la diferencia entre ambas (error), se corrigen los pesos y umbrales de las unidades de la red. Es por ello que las palabras de los conjuntos de entrenamiento y validación deben ser traducidas al formato de entrenamiento de la red, el cual contiene, además de la información de las entradas, la información de las salidas correctas para cada entrada. También es necesario traducir a dicho formato las palabras del conjunto de prueba que se utiliza para evaluar el desempeño de las redes una vez que finaliza el entrenamiento.

Formato y construcción del conjunto de entrenamiento

a) Formato del archivo de entrenamiento

Como se mencionó en la descripción de la estructura de las redes, se definió una codificación binaria única para cada posible letra de las palabras del español chileno, incluyendo el espacio. Por lo tanto, basta con traducir directamente las letras de los conjuntos de palabras para obtener las entradas de los conjuntos de entrenamiento, validación y prueba. Para obtener las salidas correspondientes a cada entrada de la red (una entrada consiste en siete caracteres adyacentes codificados), es necesario utilizar las reglas de acentuación y de traducción de texto a fonemas. Los archivos de entrenamiento están compuestos por líneas que contienen la información de la letra a traducir y su codificación para la entrada y para la salida.

Cada línea tiene el siguiente formato:

- un símbolo del alfabeto correspondiente a la letra que se desea traducir y/o acentuar.
- un caracter de tabulación que actúa como separador.
- una palabra binaria que codifica el símbolo de entrada.
- un caracter “f” que separa la codificación de la entrada y la de la salida.
- una palabra binaria que codifica la salida esperada de la red para la entrada.

La salida esperada siempre se obtiene en base al contexto de símbolos que rodean a la letra que se va a traducir. En la figura 11 se ilustra como se traduce la palabra “flechas” al formato de entrenamiento de la red. En color gris se muestra la estructura del patrón de entrenamiento de la letra “c”. Al lado izquierdo están las 7 letras de la ventana de entrada codificadas binariamente en el formato de entrada y, a la derecha, los valores esperados de las unidades de la capa de salida.

	Entrada	Salida
	abcdefghijklmnñopqrstuüvwxyz‘-	abçdefgijklmnñoprRstuxy:-’
f	00000**1**00000000000000000000000	f00000**1**00000000000000000000
l	00000000000**1**00000000000000000	f00000000000**1**00000000000000
e	0000**1**00000000000000000000000	f0000**1**00000000000000000000
c	00**1**00000000000000000000000000	f00**1**00000000000000000000000
h	0000000**1**00000000000000000000	f0000000000000000000000**1**00
a	**1**00000000000000000000000000	f**1**00000000000000000000000
s	00000000000000000**1**0000000000	f00000000000000000**1**0000000

FIGURA 11: Palabra “flechas” traducida al formato del archivo de entrenamiento de la red que traduce y acentúa. La entrada y la salida están separadas por una letra “f”. La letra “h” tiene asociada la salida de fonema nulo, representada por el símbolo “:”. En gris se muestra el patrón de entrenamiento de la letra “c”, compuesto por las 7 letras de la ventana de entrada codificadas binariamente y los valores esperados de la capa de salida.

b) Construcción del archivo de entrenamiento

El conjunto de entrenamiento se obtiene traduciendo las palabras que lo componen al formato de entrenamiento de la red, separadas entre sí por un espacio traducido también a este formato. Al conjunto traducido se le agregan al comienzo y al final tres símbolos de espacio adicionales traducidos al mismo formato. La ventana de entrada de la red va recorriendo desde el comienzo hasta el final el conjunto de entrenamiento. Como el símbolo que va a ser traducido y/o acentuado es el que se ubica en la posición central de la ventana, los tres símbolos de espacio agregados al comienzo y final del conjunto permiten que el entrenamiento comience con el caracter central de la ventana ocupado por el primer símbolo válido del conjunto, y finalice con el último (correspondientes a la letra inicial de la primera palabra y a la letra final de la última, respectivamente).

c) Algoritmo de entrenamiento en términos generales

La posición central de la ventana recorre todas las letras de las palabras del conjunto de entrenamiento. Para cada patrón, compuesto por la ventana de entrada de siete letras traducidas al formato de entrenamiento de red, se compara el valor de cada unidad de salida (S_i) de la red con el valor esperado (C_i), y se calcula el error (E_i), correspondiente a la diferencia entre dichos valores ($E_i = S_i - C_i$). Si este valor supera, para alguna salida, la cota de error máximo, se realiza la retropropagación, lo que corresponde a un entrenamiento por patrón. Para ello, se calcula el error total, que es la suma del cuadrado de todos los errores parciales ($E_{TOT} = SUM[E_i]^2$) de las unidades de la capa de salida. Utilizando este valor, se corrigen los pesos y

umbrales de las unidades de la red. Si los errores parciales en la salida no superan la cota máxima, no hay retropropagación, y se continúa con el siguiente patrón de entrenamiento, correspondiente a la entrada que se obtiene al desplazar la ventana en un caracter. Cuando se llega al último patrón, es decir, a la ventana ubicada al final del conjunto de entrenamiento, se incrementa el número de épocas por el conjunto de entrenamiento. Se realizan dos mediciones de desempeño: número de unidades de la capa de salida cuya salida difiere menos de un 10% del valor esperado, que se utiliza como cota máxima de para realizar retropropagación, y número de unidades de la capa de salida cuya salida difiere menos del 50%, utilizada para medir el número de aciertos. Este proceso se repite hasta que no se detecten errores superiores a la cota máxima para ninguna salida de cada patrón, o hasta cumplir un número máximo de épocas por el conjunto, que se fijó en 500. Después de cada época por el conjunto de entrenamiento, se realiza una época por un conjunto de validación, que es más reducido y está construido con los mismos criterios, el cual entrega una medida del rendimiento de la red para palabras no cubiertas por el conjunto de entrenamiento. Finalmente, se selecciona la red con el conjunto de pesos y umbrales para la época en la que se obtiene el mejor resultado en el conjunto de validación. Para la red seleccionada se realiza una época por un conjunto de prueba, construido con los mismos criterios que los otros dos, pero con palabras que no participaron en el entrenamiento ni en la selección de la mejor época. El desempeño de la red sobre este conjunto se utiliza como medida definitiva para evaluar la red.

En el Anexo B se hace una breve reseña a los algoritmos, ecuaciones y parámetros que rigen a las redes neuronales del tipo perceptrón de múltiples capas.

Implementación

Para el entrenamiento de las redes neuronales, la plataforma inicial de desarrollo fue el sistema operativo LINUX, con el compilador “gcc”, debido a que permite un manejo de la memoria indispensable para el desarrollo, e imposible de conseguir en DOS o Windows 3.1. Los entrenamientos finales se realizaron en Windows 98, con C++ Builder 5.0 de Borland.

El programa de entrenamiento tiene dos etapas: inicialización y entrenamiento.

a) Inicialización de la red

En la inicialización se ingresan los valores de los parámetros de la red: número de unidades de la capa de entrada, número de unidades de la capa de salida, número de capas ocultas, número de unidades de cada capa oculta, el moméntum, la tasa de aprendizaje, la cota para inicialización de valores aleatorios y la cota de error para detener el aprendizaje. Estos parámetros son descritos brevemente en el anexo B. Finalmente, los pesos y umbrales de las unidades de cada capa (excepto la de salida, que no tiene) son inicializados con valores aleatorios pequeños. Tanto los pesos como los umbrales son almacenados en memoria dinámica. Finalmente, los archivos de entrenamiento y de validación se almacenan en los arreglos de memoria estática num y num1, respectivamente.

b) Entrenamiento

Para el entrenamiento, se utilizan dos arreglos adicionales: “activacion” y “target”. El primero es para almacenar los valores de salida de cada una de las unidades de la red y el segundo para almacenar los valores esperados de las unidades de la capa de salida.

Se comienza llenando la información de las salidas de la capa de entrada, almacenada en las primeras siete filas del arreglo “num”, en las primeras posiciones del arreglo “activacion”. A

continuación, se llena el arreglo “target" con la información de la salida esperada para la capa de salida, que se encuentra en la tercera fila del arreglo “num” y corresponde a la fila central de la ventana de entrada. Después de esto, se obtiene la salida de la red, propagando las entradas hacia las capas siguientes. Con estos valores se calcula el error en cada unidad de la capa de salida, correspondiente al valor absoluto de la diferencia entre el valor esperado y el obtenido. Si para alguna salida esta diferencia es mayor que el 10%, se realiza la retropropagación del error[13].

Al final de cada época por el conjunto de entrenamiento se realiza una época por el conjunto de validación y se guarda en un archivo los siguientes valores: número de épocas, número de patrones clasificados correctamente y error cuadrático total para cada conjunto.

Cada vez que la red mejora en algún aspecto sin empeorar en otros se realiza un respaldo en archivo con los valores de los pesos y umbrales de las unidades de la red. Se considera una mejora alguno de los siguientes casos:

- disminución de los errores de clasificación en el conjunto de validación (número de salidas obtenidas que difieren menos del 50% de la salida esperada).
- dichos errores no disminuyen, pero se mantienen iguales, y disminuye el error cuadrático total del conjunto.
- ambos valores se mantienen iguales, pero disminuyen los errores en la clasificación en el conjunto de entrenamiento.

El entrenamiento se detiene cuando se cumplen 500 épocas por el conjunto.

Para la época con mejor resultado se almacenan en un archivo, para todos los patrones con salida errónea, tanto para el conjunto de entrenamiento como para el de validación y prueba los siguientes datos:

- tipo de error: diferencia entre salida esperada y salida obtenida mayor del 10% o mayor que el 50%
- número del patrón (ver figura 11) en el conjunto
- valor de la salida obtenida
- valor de la salida esperada
- error cuadrático medio total: es la suma de los errores cuadráticos medios parciales, correspondientes a la suma del cuadrado de la diferencia entre el valor esperado y el valor obtenido en cada unidad de la capa de salida para cada patrón del conjunto correspondiente (entrenamiento, validación y prueba)

Cuando finaliza el entrenamiento, se libera la memoria dinámica, y finaliza el programa.

Pruebas

Como se describió en las secciones anteriores, se entrenaron redes para tres objetivos: traducción de texto a fonemas, acentuación y traducción más acentuación.

Para cada uno de estos tres objetivos se implementaron 6 arquitecturas de red modificando el número de unidades de la capa oculta de modo de determinar la influencia del tamaño de la red en el desempeño. Luego, se utilizaron en total 18 arquitecturas de red (3 * 6). En la red más grande se utilizó el mismo número de unidades que en la red NETtalk[13] (80 unidades). En las otras cinco arquitecturas se utilizaron 30, 20, 10, 7 y 6 unidades respectivamente.

Para medir la incidencia de las condiciones iniciales en el desempeño de cada arquitectura de red, se realizaron 6 entrenamientos con distintas condiciones iniciales para cada arquitectura. Luego, el total de redes entrenadas fue 108 (3 * 6 * 6).

Para el entrenamiento de cada red se usaron tres conjuntos de palabras: entrenamiento, validación y prueba. El primero se utilizó para el entrenamiento de la red; el de validación se usó para determinar cual fue la mejor de las 500 épocas del entrenamiento; el de prueba se usó para obtener una medida del desempeño de la red, independizada del entrenamiento, para la época seleccionada.

Para medir el desempeño de las redes se utilizó el número de errores de clasificación, correspondiente al número de patrones (ver figura 11) para los cuales una o más salidas difieren más del 50% del valor esperado. El número de errores de clasificación se usó tanto para seleccionar la mejor época durante el entrenamiento como para la evaluar el desempeño, para esa época, de la red en el conjunto de prueba.

Para cada una de las 18 arquitecturas de red, se seleccionó la que presentó el mejor desempeño en el conjunto de prueba entre las seis entrenadas con distintas condiciones iniciales. En el capítulo Resultados se presenta el valor de este desempeño para las 18 redes. A su vez, para cada uno de los tres objetivos en estudio, se seleccionó la red con el mejor desempeño y el mínimo número de unidades en la capa oculta. Como resultado, se obtuvieron tres redes, una para cada tipo de objetivo. En el capítulo Resultados, para estas tres redes, se presentan para cada conjunto (entrenamiento, validación y prueba) los parámetros de desempeño descritos en la Tabla 14.

Tabla 14: Descripción de los parámetros de desempeño de las redes en los conjuntos de entrenamiento, validación y prueba, para cada objetivo en estudio (traducción a fonemas, acentuación y traducción más acentuación).

Parámetro de desempeño	Descripción
Número de Errores de Clasificación	número de patrones para los cuales una o más unidades de la capa de salida difiere en un 50% o más del valor esperado (Ejemplo: la salida esperada para la unidad que representa el sonido “a” tiene valor 1,0 y la red generó una salida 0,4)
Porcentaje de Aciertos	porcentaje de patrones sin Error de Clasificación sobre el total de patrones del conjunto
Varianza del Número de Errores de Clasificación (en el caso en que existan Errores de Clasificación)	Es la varianza de los Errores de Clasificación obtenidos en las 6 redes con la misma arquitectura, entrenadas con distintas condiciones iniciales
Error Cuadrático Medio Total	es la suma de los errores cuadráticos medios parciales, correspondientes a la suma del cuadrado de la diferencia entre el valor esperado y el valor obtenido en cada unidad de la capa de salida para cada patrón (ver figura 11) del conjunto correspondiente (entrenamiento, validación y prueba)
Varianza del Error Cuadrático Medio	Es la varianza del Error Cuadrático Medio obtenido en las 6 redes con la misma arquitectura, entrenadas con distintas condiciones iniciales
Número Total de Patrones que componen cada conjunto	Número Total de Patrones que componen el conjunto respectivo (entrenamiento, validación y prueba)
Número de la época con mejor desempeño	Número de la época para la cual se obtuvo el mejor rendimiento (menor número de errores de clasificación) en el conjunto de validación, sobre el total de 500 épocas realizadas en el conjunto de entrenamiento

Para medir la incidencia de las condiciones iniciales en desempeño de las redes, en el capítulo Resultados se presenta el número de errores de clasificación en los conjuntos de entrenamiento, validación y prueba y la varianza de este error para los seis entrenamientos realizados con las arquitecturas de las tres redes seleccionadas.

En las redes que realizan acentuación, se presentan, adicionalmente, los parámetros de desempeño descritos en la Tabla 15. Los errores de clasificación se desglosan, para cada conjunto, en errores por falta de acento (falso negativo: vocal que debió ser acentuada y no lo fue) y errores por acento indebido (falso positivo: vocal que no debía ser acentuada y si lo fue). Para ambos casos se indica el número de veces que se produjo el error en cada vocal. Finalmente, se muestra el porcentaje de palabras mal acentuadas de cada tipo (monosílabas, agudas no monosílabas, graves y esdrújulas) sobre el total de palabras del conjunto correspondiente.

Tabla 15: Descripción de los parámetros de desempeño específicos de las redes que realizan acentuación, en los conjuntos de entrenamiento, validación y prueba.

Parámetro de desempeño	Descripción
Vocales con error tipo falso negativo	vocales no acentuadas que sí debieron ser acentuadas
Vocales con error tipo falso positivo	vocales que no debieron ser acentuadas y sí lo fueron
Porcentaje de palabras mal acentuadas	el porcentaje de palabras mal acentuadas sobre el total de palabras del conjunto, separado por tipo de palabra: monosílabo, aguda no monosilábica, grave y esdrújula.

Finalmente, se hace un análisis de los resultados, en donde se explican las causas de error que se detectaron.

2.2 Asistente de lectura

Una vez analizado el tema de la traducción de texto a fonemas, se aborda a continuación el diseño de una interfaz de software que permita a usuarios no-videntes escuchar un texto en español, almacenado en un archivo, a través de una tarjeta de sonido.

Para obtener una salida audible, se utiliza una tarjeta de sonido Sound Blaster AWE32 que, además, incluye un software para síntesis de voz en inglés y español (TextAssist). Este software no está orientado a personas no videntes, por lo que es necesario diseñar una interfaz adecuada para ellos.

2.2.1 TextAssist

TextAssist es un conjunto de aplicaciones de síntesis y reconocimiento de voz desarrolladas para Windows por CREATIVE LABS para sus modelos de tarjeta de sonido posteriores al Sound Blaster 16 ASP (ver Anexo C). De todas ellas, las aplicaciones relacionadas con la síntesis de voz más relevantes son Texto'LE y TAAPI de TextAssist.

Texto'LE [14]

Esta es una aplicación que realiza síntesis de voz a partir de un texto. El texto puede ser ingresado directamente en una ventana de edición de la misma aplicación, o puede ser obtenido a partir de un archivo. Los procesos realizados sobre el texto para generar la voz que lo pronuncia son transparentes al usuario, es decir, no hay acceso a la información de fonemas, acentuación, comandos de pausa y de prosodia previos a la pronunciación. El usuario puede ejecutar, ya sea mediante el "mouse" o a través de combinaciones de teclas, las siguientes acciones sobre la pronunciación del texto: iniciarla, dejarla en pausa, reanudarla y detenerla.

Los comandos restantes que se describen no son accesibles directamente a través del teclado, es decir, sólo son modificables mediante el menú de barra o mediante el "mouse".
Texto'LE posee nueve voces diferentes para realizar la lectura. Sobre estas nueve voces se pueden efectuar las siguientes modificaciones: aumentar o disminuir el volumen, aumentar o disminuir la velocidad de pronunciación y modificar el tono de la voz.

Existen dos modalidades de pronunciación del texto: oración y palabra. En el modo oración el texto es leído en forma continua de comienzo a fin. En el modo palabra la lectura se detiene después de cada palabra.

También se puede modificar el largo de la pausa asociada a los diferentes signos de puntuación. En esta aplicación, se clasifican en dos grupos, de acuerdo al largo de la pausa: punto y coma.

Existen, además, una serie de comandos para crear texto cantado, que controlan la frecuencia y la duración de cada fonema. Mediante estos comandos, es posible realizar un control de la prosodia del texto, pero en un grado muy limitado.

TAAPI de TextAssist [15, 16]

TAAPI es una biblioteca de funciones para desarrollar aplicaciones no comerciales de síntesis de voz en Windows 3.1. Texto'LE está implementado con TAAPI.

Existen, al menos, dos versiones de TextAssist, una para el inglés y otra para el Español. El sonido de voz sintetizada de la versión en inglés es más nítido que en la versión en español, con menos ruido y mayor inteligibilidad, pero se aprecia el acento inglés en los fonemas. Por ello, para generar la salida de sonido del texto, se utilizaron ambos conjuntos de fonemas por separado, de modo de evaluar la diferencia.

El software TextAssist recibe como entrada texto traducido a fonemas con acentuación y puntuación, y produce una salida audible correspondiente a la lectura del texto original. Este software también puede recibir como entrada el texto original, realizar la traducción de texto a fonemas con acentuación, asignar las pausas y la prosodia de acuerdo a las puntuaciones del texto, y producir la salida de voz. Sin embargo, en este segundo caso no hay forma de intervenir la manera en que realizan dichos procesos sobre el texto.

Para el caso de TextAssist en inglés, el resultado de la síntesis de voz a partir de un texto sin procesar es una cadena de fonemas para el inglés. Por ello, las etapas de traducción a fonemas deben realizarse en forma independiente del software, entregando a éste sólo el resultado final del procesamiento del texto (fonemas con acentuación y puntuación), ya que las palabras entregadas son del español.

Por otra parte, el sonido resultante de la síntesis de la versión en español no corresponde a la pronunciación que se le da en Chile, ya que algunos fonemas usados en España no se utilizan en Chile (Ej.: el sonido de la 'z', la 'll' y algunos sonidos de la 'c').

2.2.2 Diseño de interfaz del Asistente de Lectura

La aplicación diseñada para apoyo a la lectura a no videntes se llama "Asistente de Lectura". Debido a que los usuarios de esta aplicación no tienen acceso a la información visual de una pantalla de computador, el Asistente de Lectura permite realizar todos los comandos para control de pronunciación y selección de archivos mediante el teclado, con mensajes de voz apropiados para facilitar el manejo.

El Asistente de Lectura tiene dos estados de operación: Control de Lectura y Sistema Operativo. En el estado de Control de Lectura se inicia, se detiene, se deja en pausa o se reanuda la lectura o pronunciación sintetizada de un archivo de texto o de fonemas. Es posible aumentar o disminuir la velocidad y el volumen de pronunciación, además de seleccionar cualquiera de las 9 distintas voces que posee el software de síntesis de la tarjeta. Se puede traducir un texto a fonemas con acentuación ya sea mediante reglas o utilizando la red neuronal.

En el estado de Sistema Operativo se pueden recorrer, para las distintas unidades de disco, los subdirectorios y los listados de archivos de cada uno, con el objeto de buscar un archivo de texto específico para ser leído. También se pueden abrir archivos de fonemas, generados anteriormente por el mismo programa. Los comandos para realizar cada acción se activan mediante el teclado, y el Asistente envía mensajes de voz para indicar las acciones que realiza el usuario.

Sistema Operativo

El Asistente de Lectura comienza en el estado de Control de Lectura. Para cambiar al estado de Sistema Operativo, se presiona la tecla 5. Al hacer esto, se emite un mensaje de voz indicando las opciones: "Abrir *archivo de texto*: tecla 1; abrir *archivo de fonemas*: tecla 2".

Este mensaje se repite cada vez que se presiona una tecla distinta a las indicadas, excepto si la tecla es ESCAPE, en cuyo caso se retorna al estado de Control de Lectura.

Un *archivo de texto* corresponde a un archivo de texto ASCII, con acentuación y signos de puntuación del español. Un *archivo de fonemas* corresponde a un archivo generado anteriormente con el Asistente de Lectura, a partir de un *archivo de texto*, en que dicho texto ha sido traducido a fonemas con acentuación y puntuación.

Si se selecciona la opción abrir archivo con la tecla 1, se almacenan en una lista todos los archivos del directorio actual que tengan la extensión "*.txt"; si se presiona la tecla 2, la extensión es "*.phm"

Si hay archivos que tienen la extensión correspondiente, el Asistente pronuncia en forma automática el primer nombre de la lista. En caso contrario, emite el mensaje: "No hay archivos coincidentes".

Las flechas verticales se utilizan para que el Asistente pronuncie el siguiente nombre en la lista (flecha abajo) o el anterior (flecha arriba). Cuando se intenta avanzar más allá del primer o último nombre de la lista, el Asistente vuelve a pronunciar el primer o último nombre, respectivamente.

Para cambiar de directorio, se utilizan las flechas horizontales. Al presionar la flecha izquierda, se cambia al directorio anterior al actual, excepto si se está en el directorio raíz, en cuyo caso, se permanece en el mismo directorio. Al presionar la flecha derecha, se genera una lista de subdirectorios, que se recorre en forma análoga a la lista de archivos. Si no hay, el Asistente envía el mensaje “No hay subdirectorios”. Para entrar a uno de los subdirectorios, se presiona la tecla ENTER una vez que se haya escuchado su nombre.

Cada vez que se cambia de directorio, se vuelve a generar la lista de archivos y a leer automáticamente el primer nombre.
Para abrir el archivo de interés, se presiona la tecla ENTER una vez que se haya escuchado su nombre. Automáticamente se sale del Sistema Operativo y se vuelve al estado de Control de Lectura.

Para cancelar la búsqueda y salir del Sistema Operativo, se presiona la tecla ESC.

Control de Lectura

El Asistente de Lectura permite tres modalidades de traducción y acentuación en forma alternativa: mediante la biblioteca de funciones TAAPI de TextAssist, mediante reglas, y mediante redes neuronales. Cada modalidad presenta algunas características diferentes que se describen a continuación y más adelante las opciones comunes a las tres.

Teclas específicas para TAAPI de TextAssist

En esta modalidad, sólo está permitida la lectura de *archivos de texto*, y no de *archivos de fonemas*. Los siguientes son los comandos que permiten el control de la lectura de textos y las teclas asociadas a cada uno:

a) Tecla ENTER: play/stop

Si hay un archivo de texto abierto, comienza la pronunciación, o se detiene si ya estaba en curso. (Análogo a las funciones PLAY/STOP de los equipos de sonido).

b) Tecla 4: cambio del modo de lectura.

Para la lectura mediante TAAPI, existen dos modos de lectura. Cada vez que se presiona esta tecla, se cambia de un modo al otro.

- Modo automático (por defecto): en este modo, el texto seleccionado es leído de corrido, del principio al fin, excepto si el usuario ejecuta un comando de pausa o de detención.
- Modo manual: en este modo, el Asistente pronuncia una palabra y se detiene, avanzando a la siguiente cada vez que el usuario presiona la tecla espacio. Como utiliza la conversión de texto a fonemas de TextAssist, no se puede utilizar con fonemas de inglés.

Al cambiar de un modo al otro, la lectura continúa desde la posición del texto en que se realizó el cambio de modo.

c) Tecla ESPACIO: control de pausas de lectura

Su función es diferente, dependiendo del modo de lectura en que se encuentre el Asistente de Lectura:

- Modo automático: si se está pronunciando un texto, detiene la pronunciación, o la reanuda si estaba detenida (análogo a la función PAUSA de los equipos de sonido).
- Modo manual: cada vez que se presiona espacio, se pronuncia la siguiente palabra del texto.

Teclas específicas para reglas y redes neuronales

a) Tecla 0: play/stop

Si hay un archivo de fonemas abierto, o se ha creado a partir de un archivo de texto (ya sea utilizando reglas o la red neuronal), realiza la misma función de la tecla ENTER, pero para el archivo de fonemas (análogo al PLAY/STOP de los equipos de sonido).

b) Tecla ESPACIO: control de pausas de lectura

Si se está pronunciando un archivo de fonemas, detiene la pronunciación, o la reanuda si estaba detenida (análogo a la tecla PAUSA de los equipos de sonido).

c) Tecla 1: traducción a fonemas mediante la red neuronal

Si hay un archivo de texto abierto, crea un archivo de fonemas, utilizando para ello la red neuronal.

d) Tecla 2: traducción a fonemas mediante reglas

Si hay un archivo de texto abierto, genera un archivo de fonemas, utilizando para ello las reglas.

Teclas de funciones comunes a todas las modalidades

a) Tecla 5: cambio de estado de operación

Cambia del estado de Control de Lectura al estado de Sistema Operativo. Como se indicó anteriormente, se activa un mensaje de voz indicando si se desea abrir un *archivo de texto* o un *archivo de fonemas*.

b) Teclas '+' y '–': control de volumen

Aumenta o disminuye el volumen de pronunciación, respectivamente, con niveles desde 0 hasta 19.

c) Tecla ESC: salir del programa

d) Flechas horizontales: selección de voz

Cambian la voz de pronunciación del texto. Hay 9 voces diferentes, y con las flechas horizontales se recorre la lista de voces en uno u otro sentido.

e) Flechas verticales: control de velocidad de pronunciación

Aumenta (flecha arriba) o disminuye (flecha abajo) la velocidad de pronunciación.

f) Tecla 3: almacena configuración

Guarda la configuración de parámetros actual: voz seleccionada, volumen, velocidad, y ruta de acceso o path al entrar al Sistema Operativo.

2.2.3 Implementación

La aplicación Asistente de Lectura fue desarrollada en C para Windows 3.1 y puede ejecutarse también en Windows 95 y 98, ya que el software que realiza la síntesis de voz, TAAPI de TextAssist [15, 16], tiene bibliotecas de desarrollo sólo para estos sistemas operativos. Las funciones que se utilizaron de esta biblioteca son descritas en el Anexo C, junto con las características principales del método de conversión de texto a voz de TextAssist. Estos sistemas operativos satisfacen los requerimientos de memoria para las distintas redes implementadas en este trabajo de título.

El Asistente de Lectura está desarrollado en Borland C 3.1 para Windows. Esto se debe a que el software de síntesis de voz de la tarjeta de sonido y el paquete de desarrollo están orientados para aplicaciones en Windows. El paquete permite desarrollar aplicaciones en lenguaje C, Pascal y Visual Basic. Se eligió el primer lenguaje de programación debido a que Visual Basic tiene más restricciones para controlar el programa, y a que se tenía mayor experiencia en programación en C que en Pascal.

El paquete de desarrollo posee funciones que pronuncian texto en español a partir de un archivo. Para ello, el propio software lleva a cabo los procesos de traducción de texto a fonemas, acentuación, y control de la prosodia de ese texto.

La razón para realizar el mismo proceso en forma alternativa es dar un sonido de lectura más cercano al acento que se le da al español en Chile. Sin embargo, esta es sólo una etapa del desarrollo, ya que aún no se ha elaborado un control de la prosodia. En el programa coexisten las tres modalidades: síntesis mediante el software de la tarjeta, síntesis a través de la traducción a fonemas y acentuación por medio de una red neuronal, y síntesis por reglas.

El programa tiene la estructura de un proyecto, que consiste en varios módulos o programas en C, agrupados de acuerdo a las funciones que realizan.

Ciclo principal del programa

El módulo principal tiene la estructura básica de cualquier programa de Windows, es decir, el flujo del programa se controla mediante una cola de mensajes que el sistema operativo envía al programa. Como es una aplicación para no videntes, los mensajes que se utilizan son los del teclado, los del software generador de voz, y los de control de flujo para inicializar y terminar el programa. No se usan menús de barras ni mensajes del “mouse”. Existen dos estados de operación en el Asistente de Lectura: Control de Lectura y Sistema Operativo. Algunos mensajes de Windows se utilizan en ambos estados, pero las acciones son diferentes para cada uno (por ejemplo, en el modo Sistema Operativo, durante el proceso de búsqueda de un archivo que se desea abrir, las flechas verticales permiten recorrer los archivos de un directorio

pronunciando sus nombres y, en Control de Lectura, las flechas controlan la velocidad de pronunciación de la lectura de un archivo). Por ello, existe una variable que indica en cual de los dos estados se encuentra el programa. A continuación se describe las acciones que ejecuta el Asistente de Lectura ante los principales mensajes que envía el sistema operativo Windows.

Mensaje WM_CREATE

Este mensaje lo envía Windows cuando comienza la aplicación.

El Asistente abre el archivo de configuración, en donde están almacenados los parámetros de lectura: voz, volumen, número de palabras por minuto y ruta de acceso (directorio de búsqueda por defecto). Si no se encuentra este archivo, se asignan valores por defecto. Se inicializa la red neuronal que traduce y acentúa. Para ello, se lee el archivo obtenido del entrenamiento, que tiene la información de los parámetros de la red y los valores de los pesos y umbrales de las unidades correspondientes. A continuación, se asigna memoria dinámica para almacenar la información y la estructura de la red neuronal, y los pesos y umbrales son inicializados con los valores del archivo. Antes de comenzar la inicialización y una vez terminada se envían mensajes de voz que lo señalan. La variable que establece el estado de operación se inicializa para el estado de Control de Lectura, y la variable que controla el modo de lectura se inicializa para el *modo automático*.

Mensaje WM_KEYDOWN

Este mensaje se recibe cuando se ha presionado alguna tecla. Las acciones que realiza el Asistente de Lectura ante cada tecla dependen de la variable que establece el estado de operación (Control de Lectura o Sistema Operativo) y están descritas al final de la sección 2.2.2.

Si en el estado de Control de Lectura se presiona la tecla 5, la variable que establece el estado de operación cambia al estado Sistema Operativo.

Si en el estado de Sistema Operativo se presiona la tecla ESC (cancelar) o se selecciona un archivo, la variable que establece el estado de operación cambia al estado Control de Lectura.

Mensaje MsgCtsCallBack

Este mensaje es enviado por el software TAAPI [15] de TextAssist cuando ha terminado de pronunciar el texto que tiene almacenado en la cola de espera (si se encuentra en *modo automático*), o cuando ha terminado de pronunciar una palabra del texto (si se encuentra en *modo manual*). Se utiliza para controlar el número de mensajes en espera de ser pronunciados, tanto en el Control de Lectura como en el Sistema Operativo. Como se indicó al comienzo de esta sección, el Asistente de Lectura utiliza funciones de un paquete de desarrollo de CREATIVE LABS, los fabricantes de las tarjetas de sonido Sound Blaster. Estas funciones forman parte de un archivo DLL (biblioteca de enlace dinámico), y se ejecutan en forma simultánea pero independiente a la aplicación que las utiliza. Por ello, la manera de comunicarse es enviando mensajes que entran al ciclo de la aplicación.

Mensaje WM_CLOSE

Este mensaje se genera cuando el usuario sale del programa. A continuación, se libera la memoria que ha sido asignada a la aplicación y, si hay algún canal de voz abierto, se ejecuta la función de cierre de este.

Mensaje WM_DESTROY

Este mensaje se genera después de WM_CLOSE, y cierra la aplicación.

Esta es la estructura del ciclo principal del programa. A continuación se describen los módulos y funciones en C implementados para la traducción de texto a fonemas más acentuación mediante redes neuronales y reglas ortográficas.

Traducción del texto al formato de entrada de red

El primer paso es almacenar el texto a traducir en un arreglo en memoria. Este texto contiene letras del alfabeto español, espacios, números y signos de puntuación. La red sólo es capaz de traducir letras del alfabeto, por lo que es necesario filtrar el resto de la información (números, signos de puntuación, y otros caracteres). Para ello, se leen uno a uno los caracteres del arreglo, hasta encontrar un caracter que pertenezca al alfabeto, el cual es considerado como la primera letra de una palabra. Se continúa la lectura hasta encontrar un caracter que no pertenezca al alfabeto, y se considera que la palabra termina en el caracter anterior. A continuación, se realiza la traducción de esa palabra al formato de entrada de red, usando para ello las reglas de traducción de texto a fonemas y de acentuación, y se va almacenando en un archivo.

Algoritmos de red neuronal

Los algoritmos de la red neuronal son los mismos que se utilizan para el entrenamiento de la red, con la diferencia de que el archivo de entrada no contiene información de la salida esperada de la red.

Producto de las restricciones del modelo de memoria de Windows 3.1, y para evitar las limitaciones de tamaño del texto de entrada, la matriz que almacena los datos del archivo con la entrada de la red tiene un tamaño variable, pero acotado. Si el archivo de entrada de red es de mayor envergadura que la cota máxima de la matriz, ésta se va llenando por partes, hasta que se hayan obtenido las salidas para todo el archivo. Cada salida de la red es traducida al fonema correspondiente y almacenada en un archivo de fonemas; si la unidad de acentuación está activa en la capa de salida de la red, se agrega el comando de acentuación antes del fonema.

Además de traducir el texto a fonemas con acentuación, los signos de puntuación deben volver a situarse en la posición que tenían en el texto original. Para ello, se utiliza el arreglo en memoria que contiene el texto original, el cual se va recorriendo caracter a caracter. Mientras no se encuentren caracteres pertenecientes a una palabra (letras del alfabeto español, incluyendo mayúsculas, minúsculas, vocales acentuadas y la letra “ü”), se considera que los caracteres leídos corresponden a signos de puntuación, números o caracteres no válidos.

El software TAAPI, que realiza la síntesis a partir de los fonemas, tiene dos tipos de pausas para las puntuaciones. Por esto, es necesario clasificar cada signo de puntuación el alguno de estos dos tipos y luego grabarlo en el archivo de fonemas.

Pausa de coma: es una pausa corta (configurable entre 40 y 30000 ms.), durante la cual no se emite sonido. Dentro de ésta se clasifican los siguientes signos de puntuación: coma, punto y coma, dos puntos, guión.

Pausa de punto: es una pausa larga (configurable entre 380 y 30000 ms.). Dentro de ésta se clasifican los siguientes signos de puntuación: punto, signos de interrogación y exclamación.

Para que el software TAAPI interprete que el texto que se le entrega es una cadena de fonemas con acentuación, es necesario agregar ciertos formatos y palabras clave establecidos por dicho software. Si algún trozo del texto no tiene el formato de fonemas, el propio software TAAPI va a realizar la traducción de ese trozo a fonemas con acentuación automáticamente. Aprovechando esta característica, los números se dejan fuera del formato de fonemas, y son procesados por TAAPI cada vez que los encuentra.

Finalmente, los caracteres no válidos se traducen como caracteres espacio, que no afectan en nada a la pronunciación.

Traducción del texto a fonemas con acentuación mediante reglas

El procedimiento está descrito anteriormente en forma breve al final de la sección 2.1.1, en la subsección “Implementación”. El texto a traducir se encuentra en un arreglo en memoria (el mismo que se utiliza para generar la entrada de la red neuronal), y el resultado de la traducción es almacenado en un archivo de fonemas temporal.

A las reglas de traducción a fonemas con acentuación se incorporan, además, las de clasificación de puntuación descritas para la red neuronal al final de la subsección “Algoritmos de red neuronal”, en esta misma sección. El archivo se recorre linealmente de comienzo a fin, pero las reglas de traducción se aplican por palabras y las de puntuación al resto de los caracteres.

2.2.4 Pruebas

Para evaluar la aplicación Asistente de Lectura se realizaron secuencias de prueba a un grupo de 10 personas voluntarias a quienes se les vendaron los ojos. A partir de los resultados de estas pruebas se midieron aspectos de funcionalidad y de inteligibilidad del Asistente.

La secuencia de pruebas consistió en los siguientes pasos:

- abrir un archivo de texto
- traducir el archivo a fonemas
- guardar el archivo traducido
- escuchar el texto traducido a fonemas, controlando los parámetros de pronunciación (volumen, velocidad y selección de distintas voces) y los modos de lectura (de corrido o por palabra)

Como resultado de estas pruebas se realizaron las evaluaciones de funcionalidad e inteligibilidad que se describen a continuación.

Funcionalidad

Una vez que los 10 usuarios estuvieron familiarizados con los comandos de control de la aplicación a través del teclado, fueron encuestados respecto de la dificultad en su uso, clasificando el manejo de la aplicación en las siguientes categorías: FÁCIL, DIFÍCIL o REGULAR.

Inteligibilidad

Para medir la inteligibilidad del sonido de voz generado en la lectura sintetizada, los usuarios escucharon dos textos de prueba compuestos de 200 palabras, uno con frases completas y otro con palabras sueltas, con los fonemas en español y con fonemas del inglés y del español adaptados al español chileno. Como medida de evaluación de la inteligibilidad, se registró el promedio de palabras reconocidas por los usuarios la primera y la tercera vez que escucharon cada texto.

Evaluación de las características del sonido de voz

Los usuarios fueron encuestados respecto de sus percepciones personales al escuchar el sonido de voz generado por el Asistente de Lectura. Se consideraron tres categorías: gusto (agradable o desagradable), naturalidad (sonido natural o artificial) y nitidez del sonido (bueno, regular o malo).

3. RESULTADOS

En el presente capítulo se presentan los resultados obtenidos en la traducción de texto a fonemas más acentuación mediante los dos métodos descritos en este trabajo: reglas ortográficas y redes neuronales. Se realiza un análisis de los resultados, indicando bajo qué condiciones se obtuvo una acentuación y traducción de texto a fonemas de modo exitoso o incorrecto y, en caso de ser incorrecto, indicando las posibles causas de estos errores.

Finalmente, se presentan los resultados de las evaluaciones realizadas a la aplicación Asistente de Lectura desarrollada como apoyo a la lectura de no videntes.

3.1 Acentuación y traducción de texto a fonemas mediante reglas

En la Tabla 16 se presenta el tiempo de procesamiento y el porcentaje de aciertos en la traducción de texto a fonemas y en la acentuación de las 1.491 palabras que componen el conjunto de entrenamiento, utilizando los algoritmos de reglas implementados en la aplicación Asistente de Lectura, descritos en la sección 2.1.1. Estas pruebas fueron realizadas en un computador con procesador Pentium II de 350MHz, con 288 MB de memoria RAM, con sistema operativo Windows 98.

Tabla 16: Tiempo de procesamiento y porcentaje de aciertos en la traducción y acentuación del conjunto de entrenamiento mediante reglas.

Proceso	Aciertos	Tiempo [ms]
Acentuación	100%	50
Traducción a fonemas	100%	60
Total	100%	110

En la Tabla 17 se presenta el número de caracteres de los conjuntos de entrenamiento, validación y prueba antes y después de ser traducidos a fonemas con acentuación.

Tabla 17: Número de caracteres de cada conjunto antes y después de ser traducidos a fonemas mediante reglas.

Conjunto	Número de caracteres		Porcentaje de disminución
	Antes	Después	
Entrenamiento	10025	9693	3,3%
Validación	821	798	2,8%
Prueba	867	815	6,0%

El número de caracteres cambia como consecuencia del proceso de traducción a fonemas, puesto que se eliminan los símbolos que no tienen un sonido asociado, como por ejemplo la letra “h”, y se agregan caracteres cuando aparece la letra “x”, que se traduce a las letras “ks”. Sin embargo, el número de letras que se eliminan es, por lo general, mayor y por lo tanto, el tamaño del texto traducido disminuye.

3.2 Acentuación y traducción de texto a fonemas mediante redes neuronales

A continuación se presentan los resultados obtenidos en las redes neuronales para los tres objetivos en estudio: red para **traducción a fonemas**, red para **acentuación** y red para **traducción de texto a fonemas más acentuación**.

Como se describió en la subsección “Pruebas”, al final de la sección 2.1.2, para cada uno de estos 3 objetivos se implementaron 6 arquitecturas de red (18 en total) modificando el número de unidades de la capa oculta (80, 30, 20, 10, 7 y 6 unidades) de modo de determinar la influencia del tamaño de la red en el desempeño. Para las 18 arquitecturas de red se realizaron 6 entrenamientos de 500 épocas cada uno con distintas condiciones iniciales, con el objeto de medir la incidencia de éstas en el desempeño de cada arquitectura. De estos 6 entrenamientos, se seleccionó la red con mejor desempeño en el conjunto de prueba, correspondiente a la que tiene el menor número de errores de clasificación (ver Tabla 14, en capítulo Metodología) en las 500 épocas, obteniéndose la mejor red para cada una de las arquitecturas.

En los resultados, se muestra el número de errores de clasificación en el conjunto de prueba para las 18 arquitecturas de red, separado por objetivo. Con ello se determinó la arquitectura de red (y por lo tanto el tamaño de la capa oculta) con mejor desempeño en el conjunto de prueba por cada objetivo, obteniendo tres redes.

Para estas tres redes se presentan los parámetros de desempeño descritos en la Tabla 14 del capítulo Metodología, obtenidos en cada conjunto (entrenamiento, validación y prueba).

Para estas tres arquitecturas de red seleccionadas se presenta el número de errores de clasificación que se obtuvo en los conjuntos de entrenamiento, validación y prueba junto con la varianza de este error para los 6 entrenamientos con distintas condiciones iniciales, con el objeto de determinar el efecto de éstas en el desempeño de las redes.

Para las redes que realizan acentuación se presentan, además, los parámetros descritos en la Tabla 15 del capítulo Metodología.

Se realizó una comparación del desempeño en traducción a fonemas entre la red que sólo traduce y la que traduce y acentúa; de igual modo, se comparó el desempeño en la acentuación entre la red que sólo acentúa y la que traduce y acentúa, utilizando como criterio para ambos casos el desempeño en los distintos conjuntos.

Selección de la mejor arquitectura de red para cada objetivo

En la Tabla 18 se muestra para cada objetivo el desempeño de las 6 arquitecturas de red implementadas, medido como el menor número de errores de clasificación en el conjunto de prueba para las 500 épocas de entrenamiento, obtenido a partir de la red con mejor desempeño de los 6 entrenamientos con distintas condiciones iniciales realizados en cada arquitectura.

Las arquitecturas de red que realizan acentuación tienen su mejor desempeño con 30 unidades. Las arquitecturas de red que sólo traducen a fonemas comienzan a presentar errores con 6 o menos unidades en la capa oculta, por lo que se seleccionó la red con 7 unidades en dicha capa.

Tabla 18: Número de errores de clasificación en el conjunto de prueba de acuerdo al número de unidades de la capa oculta, separado por objetivo, obtenido de la red con mejor desempeño los 6 entrenamientos con distintas condiciones iniciales realizados en cada arquitectura de red

Desempeño de las redes para cada arquitectura			
Número de unidades capa oculta	**Traducción**	**Acentuación**	**Traducción más acentuación**
80	0	37	43
30	0	37	42
20	0	40	50
10	0	43	53
7	0	47	57
6	69	53	127

Selección del entrenamiento con mejor desempeño para la arquitectura de red seleccionada al cambiar las condiciones iniciales

En las Tablas 19 y 20 se muestra para las redes de acentuación y de traducción más acentuación el desempeño de los 6 entrenamientos con distintas condiciones iniciales realizados con las arquitecturas seleccionadas en el punto anterior que, en ambos casos, tiene 30 unidades en la capa oculta. Se presenta para los conjuntos de entrenamiento, validación y prueba el número de errores de clasificación y la varianza del error obtenida a partir de los 6 entrenamientos.

Tabla 19: Desempeño de los seis entrenamientos realizados con distintas condiciones iniciales a la red de acentuación con 30 unidades en la capa oculta. Se presenta el número de errores de clasificación en el conjunto de prueba y la varianza del error para los seis entrenamientos.

Efecto de las condiciones iniciales en la red que sólo acentúa con 30 unidades en la capa oculta			
Condiciones iniciales del Entrenamiento	**Número de errores de Clasificación**		
	Entrenamiento	**Validación**	**Prueba**
CI 1	39	25	43
CI 2	26	25	46
CI 3	8	26	43
CI 4	7	24	37
CI 5	15	27	39
CI 6	9	27	37
Varianza	162,7	1,5	13,8

La red para acentuación obtenida en el cuarto entrenamiento es la que tuvo mejor desempeño.

Tabla 20: Desempeño de los seis entrenamientos realizados con distintas condiciones iniciales a la red de traducción más acentuación con 30 unidades en la capa oculta. Se presenta el número de errores de clasificación en el conjunto de prueba y la varianza del error para los seis entrenamientos.

Efecto de las condiciones iniciales en la red que traduce y acentúa con 30 unidades en la capa oculta			
Condiciones iniciales del Entrenamiento	Número de errores de Clasificación		
	Entrenamiento	Validación	Prueba
CI 1	10	28	46
CI 2	13	31	44
CI 3	60	28	42
CI 4	15	22	44
CI 5	33	24	46
CI 6	9	20	42
Varianza	399,5	17,5	3,2

La red para traducción a fonemas más acentuación obtenida en el sexto entrenamiento es la que tuvo mejor desempeño.

La red que sólo traduce a fonemas no presentó errores de clasificación, por lo que no se hizo el cálculo de la varianza de este error para los seis entrenamientos.

Desempeño de la red seleccionada para traducción de texto a fonemas

La Tabla 21 muestra los valores obtenidos en los parámetros de desempeño para la red de traducción de texto a fonemas con 7 unidades en la capa oculta. Se indica para los conjuntos de entrenamiento, validación y prueba, el número de errores de clasificación, el porcentaje de aciertos, el error cuadrático medio, la varianza del error cuadrático medio (para de las 6 redes entrenadas con distintas condiciones iniciales), el número de época en la que se obtuvo el mejor rendimiento en el conjunto de validación sobre el total de 500 épocas, y el número total de patrones del conjunto.

Tabla 21: Parámetros de desempeño de la red de traducción de texto a fonemas con 7 unidades en la capa oculta.

Conjunto	Error de Clasificación (Error ≥ 50 %)	Aciertos %	Error cuadrático medio (E2)	Varianza del E2	Época	Total de patrones
Entrenamiento	0	100%	75,2	18,7	102	11515
Validación	0	100%	6,53	0,2		955
Prueba	0	100%	8,14	0,1		1001

No se produjeron errores de clasificación en ninguno de los conjuntos (entrenamiento, validación y prueba).

Desempeño de la red seleccionada para acentuación

La Tabla 22 muestra los parámetros de desempeño de la red de acentuación con 30 unidades en la capa oculta. Se indica para los conjuntos de entrenamiento, validación y prueba, el número de errores de clasificación, el porcentaje de aciertos, la varianza de los errores de clasificación (para de las 6 redes entrenadas con distintas condiciones iniciales), el error cuadrático medio, la varianza del error cuadrático medio (para de las 6 redes), el número de época en la que se obtuvo el mejor rendimiento en el conjunto de validación sobre el total de 500 épocas realizadas, y el número total de patrones del conjunto.

Tabla 22: Parámetros de desempeño en las redes de traducción de texto a fonemas más acentuación con 30 unidades en la capa oculta.

Conjunto	Error de Clasificación (Error ≥ 50 %)	Aciertos %	Varianza de los Errores de Clasificación	Error cuadrático medio (E2)	Varianza del E2	Época	Total de patrones
Entrenamiento	7	99,94%	162,7	9,13	157,0	468	11515
Validación	24	97,49%	1,5	21,90	1,2		955
Prueba	37	96,30%	13,8	32,78	11,4		1001

Las Tablas 23, 24 y 25 muestran los errores de acentuación en los conjuntos de prueba, validación y entrenamiento, respectivamente, para red de acentuación. Los errores están desglosados por tipo de error en la acentuación de vocales: falso negativo y falso positivo y, para cada tipo, se presenta además el porcentaje de palabras mal acentuadas sobre el total de palabras del conjunto correspondiente, separado por tipo de palabra: monosílabo, aguda no monosilábica, grave y esdrújula.

Tabla 23: Desglose de errores de clasificación en la acentuación del conjunto de prueba para la red de acentuación: número de errores de acuerdo al tipo de vocal mal acentuada (falso negativo y falso positivo); porcentaje de errores de acuerdo al tipo de palabra, sobre el total de palabras del conjunto

	Acentuación (Total palabras: 135)									
	Error de Clasificación (Error ≥ 50 %)									
	Falso negativo					**Falso positivo**				
	à	**è**	**ì**	**ò**	**ù**	**a**	**e**	**i**	**o**	**u**
	8	3	6	6	5	4	5	0	1	3
Monosílabos (0,0%)	0,0%					0,0%				
Agudas (3,0%)	0,0%					3,0%				
Graves (27,4%)	20,7%					6,7%				
Esdrújulas (0,0%)	0,0%					0,0%				
Total (30,4%)	**20,7%**					**9,7%**				

Tabla 24: Desglose de errores de clasificación en la acentuación del conjunto de validación para la red de acentuación

	Acentuación (Total palabras: 135)									
	Error de Clasificación (Error ≥ 50 %)									
	Falso negativo					Falso positivo				
	à	è	ì	ò	ù	a	e	i	o	u
	5	3	3	4	2	2	2	0	0	3
Monosílabos (0,7%)	0,7%					0,0%				
Agudas (3,0%)	0,0%					3,0%				
Graves (14,1%)	11,9%					2,2%				
Esdrújulas (0,0%)	0,0%					0,0%				
Total (17,8%)	**12,6%**					**5,2%**				

Tabla 25: Desglose de errores de clasificación en la acentuación del conjunto de entrenamiento para la red de acentuación

	Acentuación (Total palabras: 1491)									
	Error de Clasificación (Error ≥ 50 %)									
	Falso negativo					Falso positivo				
	à	è	ì	ò	ù	a	e	i	o	u
	2	0	2	0	0	0	0	1	0	1
Monosílabos (0,0%)	0,0%					0,0%				
Agudas (0,1%)	0,0%					0,1%				
Graves (0,4%)	0,3%					0,1%				
Esdrújulas (0,0%)	0,0%					0,0%				
Total (0,5%)	**0,3%**					**0,2%**				

Desempeño de la red seleccionada para traducción de texto a fonemas y acentuación

La Tabla 26 muestra los parámetros de desempeño de la red de traducción a fonemas más acentuación con 30 unidades en la capa oculta. Se indica para los conjuntos de entrenamiento, validación y prueba, el número de errores de clasificación, el porcentaje de aciertos, la varianza de los errores de clasificación (para de las 6 redes entrenadas con distintas condiciones iniciales), el error cuadrático medio, la varianza del error cuadrático medio (para de las 6 redes), el número de época en la que se obtuvo el mejor rendimiento en el conjunto de validación sobre el total de 500 épocas realizadas, y el número total de patrones del conjunto.

Tabla 26: Parámetros de desempeño en las redes de traducción de texto a fonemas más acentuación con 30 unidades en la capa oculta.

Conjunto	Error de Clasificación (Error ≥ 50 %)	Aciertos %	Varianza de los Errores de Clasificación	Error cuadrático medio (E2)	Varianza del E2	Época	Total de patrones
Entrenamiento	9	99,9%	399,5	36,61	299,1	264	11515
Validación	20	97,9%	17,5	21,98	33,8		955
Prueba	42	95,8%	0,1	37,22	46,6		1001

No se produjeron errores en la traducción de texto a fonemas. Las Tablas 27, 28 y 29 muestran los errores de acentuación en los conjuntos de prueba, validación y entrenamiento, respectivamente. Los errores están desglosados por tipo de error en la acentuación de vocales: falso negativo y falso positivo y, para cada tipo, se presenta además el porcentaje de palabras mal acentuadas sobre el total de palabras del conjunto correspondiente, separado por tipo de palabra: monosílabo, aguda no monosilábica, grave y esdrújula.

Tabla 27: Desglose de errores de clasificación en la acentuación del conjunto de prueba para la red de traducción a fonemas más acentuación: número de errores de acuerdo al tipo de vocal mal acentuada (falso negativo y falso positivo); porcentaje de errores de acuerdo al tipo de palabra, sobre el total de palabras del conjunto

	Acentuación (Total palabras: 135)									
	Error de Clasificación (Error ≥ 50 %)									
	Falso negativo					**Falso positivo**				
	à	**è**	**ì**	**ò**	**ù**	**a**	**e**	**i**	**o**	**u**
	5	3	6	4	4	2	5	6	0	3
Monosílabos (0,0%)	0,0%					0,0%				
Agudas (3,7%)	0,0%					3,7%				
Graves (23,0%)	16,3%					6,7%				
Esdrújulas (1,5%)	0,0%					1,5%				
Total (28,2%)	**16,3%**					**11,9%**				

Tabla 28: Desglose de errores de clasificación en la acentuación del conjunto de validación para la red de traducción a fonemas más acentuación

	Acentuación (Total palabras: 135)									
	Error de Clasificación (Error ≥ 50 %)									
	Falso negativo					**Falso positivo**				
	à	**è**	**ì**	**ò**	**ù**	**a**	**e**	**i**	**o**	**u**
	4	4	1	5	1	2	1	0	0	2
Monosílabos (0,7%)	0,7%					0,0%				
Agudas (2,2%)	0,7%					1,5%				
Graves (11,8%)	9,6%					2,2%				
Esdrújulas (0,0%)	0,0%					0,0%				
Total (14,7%)	**11,0%**					**3,7%**				

Tabla 29: Desglose de errores de clasificación en la acentuación del conjunto de entrenamiento para la red de traducción a fonemas más acentuación

	Acentuación (Total palabras: 1491)									
	Error de Clasificación (Error ≥ 50 %)									
	Falso negativo					**Falso positivo**				
	à	**è**	**ì**	**ò**	**ù**	**a**	**e**	**i**	**o**	**u**
	2	0	2	1	1	0	0	2	1	0
Monosílabos (0,1%)	0,1%					0,0%				
Agudas (0,1%)	0,0%					0,1%				
Graves (0,4%)	0,3%					0,1%				
Esdrújulas (0,0%)	0,0%					0,0%				
Total (0,6%)	**0,4%**					**0,2%**				

3.2.3 Análisis de los resultados

El análisis de los resultados obtenidos para las distintas redes entrenadas busca seleccionar la mejor arquitectura de red para acentuación y para traducción a fonemas dentro de las tres arquitecturas implementadas, además de encontrar las causas posibles para los errores que se presentaron. Para ello se comparan los tiempos de procesamiento y los parámetros de desempeño de las redes obtenidas en cada arquitectura.

Tiempos de procesamiento

En la Tabla 30 se muestran los tiempos requeridos para traducir las palabras del conjunto de entrenamiento primero al formato de entrada de la red y luego para generar las salidas, utilizando las redes seleccionadas para cada objetivo.

Tabla 30: Tiempo para generar la salida de las redes al recibir como entrada las palabras del conjunto de entrenamiento.

Tiempos de procesamiento de las redes seleccionadas			
Etapa	**Traducción y Acentuación [s]**	**Traducción [s]**	**Acentuación [s]**
Inicialización de la red	0,2	0,2	0,2
Traducción al formato de entrada de red	0,2	0,1	0,2
Traducción a fonemas más acentuación	67,8	20,7	54,8
Total	68,2	21,0	55,2

El tiempo total de procesamiento es aproximadamente 620 veces el tiempo empleado por el sistema de reglas en el mismo procedimiento.

Mejor red en traducción a fonemas y mejor red en acentuación

A continuación se compara por separado el desempeño de las redes que traducen texto a fonemas y las redes que realizan acentuación.

Mejor red para traducción de texto a fonemas

Tanto la red que sólo traduce a fonemas como la que traduce y acentúa no presentaron errores de traducción a fonemas en ninguno de los tres conjuntos de palabras, y sólo tuvieron diferencias en el error cuadrático medio total y el tiempo de procesamiento de cada conjunto. El error cuadrático medio no es comparable entre ambas redes, ya que la que traduce y acentúa incorpora los errores de acentuación en la medición del error cuadrático medio. Al traducir a fonemas el conjunto de entrenamiento, la diferencia en tiempo de procesamiento es de 47,2 segundos a favor de la red que sólo traduce, debido a que tiene menos unidades en la capa oculta y una unidad menos en la capa de salida. Luego, si sólo se requiere traducir a fonemas, la mejor red es la que sólo traduce a fonemas.

La red que traduce y acentúa no empeoró su desempeño en traducción a fonemas por el hecho de realizar simultáneamente acentuación.

Mejor red para acentuación

En la Tabla 31 se presenta resumido el desempeño de las redes en la acentuación, correspondiente al número de errores de clasificación en los conjuntos de prueba, validación y entrenamiento.

Tabla 31: Errores de clasificación en la acentuación en los conjuntos de entrenamiento, validación y prueba sobre el total de patrones de cada conjunto para la red que traduce y acentúa y la que sólo acentúa.

Errores de Clasificación en acentuación			
	Traducción más acentuación	**Acentuación**	**Total patrones**
Conjunto de prueba	42	37	1001
Conjunto de validación	20	24	955
Conjunto de entrenamiento	9	7	11515

No se puede determinar que alguna de las dos redes sea mejor en la acentuación. El número de errores en el conjunto de prueba y entrenamiento es mayor en la red que traduce y acentúa, mientras que para el conjunto de validación esta misma red presenta un error menor que la que sólo traduce. Como se aprecia en las Tablas 19 y 20, el efecto de las condiciones iniciales en el desempeño de las arquitecturas de red seleccionadas es mayor que la diferencia de desempeño que presentan al compararlas, por lo que no se puede indicar que una red sea mejor que la otra.

En todas las redes para la acentuación de palabras, a diferencia del caso de traducción a fonemas, se presentaron errores de clasificación en el conjunto de entrenamiento, es decir, un porcentaje de letras para las cuales la red entrega una salida incorrecta.

Incidencia de las condiciones iniciales

Como se muestra en las Tablas 19 y 20, el efecto de las condiciones iniciales en el desempeño de las redes que acentúan y las que traducen y acentúan es muy alto, presentando una varianza superior a 160 en el número de errores de clasificación en los conjuntos de entrenamiento. Al comparar el desempeño en la acentuación entre la red seleccionada para traducción y acentuación y la que sólo traduce, la diferencia es inferior a la diferencia de desempeño entre redes de la misma arquitectura variando las condiciones iniciales.

Análisis de errores

La traducción de texto a fonemas no presentó errores de clasificación en las redes ya entrenadas. Por ello, sólo se analizan los errores en la acentuación de acuerdo a dos criterios: tipo de palabra mal acentuada (monosílabo, aguda, grave y esdrújula) y tipo de error (falso positivo o acentuación indebida y falso negativo o falta de acentuación). Finalmente, se analizan las causas detectadas para algunos errores en la acentuación.

Distribución de errores de acentuación de acuerdo al tipo de palabra

En la Tabla 32 se presentan resumidos los porcentajes de palabras con error de clasificación en la acentuación sobre el total de palabras de los conjuntos de entrenamiento, validación y prueba, para las redes que acentúan, distribuidos por tipo de palabra.

Tabla 32: Porcentaje de palabras con error de de clasificación en la acentuación sobre el total de palabras de los conjuntos de entrenamiento (11515), validación (955) y prueba (1001), para la red que traduce y acentúa y la que sólo acentúa, distribuidos por tipo de palabra.

Porcentaje de errores en la acentuación						
	Sólo Acentuación			Traducción más acentuación		
	Prueba	Validación	Entrenamiento	Prueba	Validación	Entrenamiento
Monosílabos	0,0%	0,7%	0,0%	0,0%	0,7%	0,1%
Agudas	3,0%	3,0%	0,1%	3,7%	2,2%	0,1%
Graves	27,4%	14,1%	0,4%	23,0%	11,8%	0,4%
Esdrújulas	0,0%	0,0%	0,0%	1,5%	0,0%	0,0%
Total	30.4%	17,8%	0,5%	28,2%	14,7%	0,6%

Las dos redes que realizan acentuación presentaron, para los conjuntos de entrenamiento, validación y prueba, el mayor porcentaje de error en las palabras graves. Las palabras agudas ocupan el segundo mayor porcentaje de errores. Los monosílabos presentaron el tercer porcentaje de error. Las palabras esdrújulas presentaron el menor porcentaje de errores (sólo dos casos en el conjunto de prueba de la red que traduce y acentúa).

Distribución de errores de acentuación de acuerdo al tipo de error: falso positivo o falso negativo

A partir de los resultados expuestos en las Tablas 23, 24, 25, 27, 28 y 29 se ve que en las redes que acentúan, para los conjuntos de entrenamiento, validación y prueba, las palabras agudas y esdrújulas presentan un mayor porcentaje de error de tipo falso positivo (palabras acentuadas en vocales que no llevan acento), mientras que las graves y monosílabos presentan un mayor porcentaje de error de tipo falso negativo (palabras que no fueron acentuadas en vocales que llevan acento).

En las mismas Tablas, al observar los tipos de errores de desglosados por vocal, no se encontró una regularidad en la distribución del error en las dos redes para los tres conjuntos.

Causas de error detectadas

Para analizar los errores que se presentaron en la acentuación, en la Tabla 33 se muestra una lista de ejemplos con palabras de los distintos conjuntos en las que se produjo error de clasificación en las dos redes de acentuación. Para cada palabra se muestra en negrilla la letra en la que se presentó el error y, centrada en torno a ella, se dibuja una ventana de 7 letras, que corresponde a la entrada de la red para la letra con error en la acentuación. Si existe alguna palabra del conjunto de entrenamiento que puede tener relación directa con el error, se escribe a la derecha de la palabra errónea. Los caracteres “*” representan las letras de las palabras adyacentes para los casos en que hay más de una palabra en la ventana de entrada. Analizando las palabras mal acentuadas se detectaron dos fuentes de error que se describen a continuación de las tablas.

Tabla 33: Ejemplos de errores de acentuación en los distintos conjuntos.

Palabras mal acentuadas		Palabra relacionada	Otro ejemplo
	- c h **i** n c h o r r o	chinche	
	- g u **i** l l a	guillatún	
	- c h **a** n c h u l l o	chancho	
	* * - **i** t r i o	árbitro	Vitrificar
	j o f **a** i n a	vainilla	
	m a n **a** g u a	enagüetas	
	m e m **o** r i a	memorándum	Memorial
	* - y **e** s c a	yesquero	Refrescar

a) Palabras distintas pero con segmentos de ellas iguales o similares

Para realizar la acentuación, la información de entrada de la red es una ventana de 7 letras. Por ejemplo, al considerar dicha ventana para la salida errónea de las palabras “guilla” y “guillatún”, se tiene lo siguiente:

```
1234567
-guilla          "i" acentuada
-guillatún       "i" no acentuada
```

Lo mismo ocurre con la palabra “chinche” del conjunto de validación, pues dentro del conjunto de entrenamiento también se encuentra la palabra “chinchorro” y se produce la misma situación anterior.

```
1234567
-chinche         "i" acentuada
-chinchorro      "i" no acentuada
```

En la palabra “guilla”, la letra “i” tiene acento implícito y, por lo tanto, la salida de la red debería activarse. Por otra parte, en la palabra “guillatún” la letra “i” no tiene acento y, luego, la salida de la red no debería activarse. Por lo tanto, para una misma entrada la red debe generar salidas opuestas. En el primer ejemplo, el sistema acentuó tanto la letra “i” como la letra “ú” de la palabra “guillatún”. Este problema se produce debido a que la palabra que se va a acentuar no cabe íntegramente dentro de la ventana de entrada de 7 letras de la red y de esta manera se puede presentar para más de una palabra una misma entrada o entradas similares con salidas de acentuación opuestas dentro del conjunto de entrenamiento. Si este segmento pertenece a una palabra grave y coincide con la estructura de una aguda, o viceversa, la red recibirá información contradictoria durante su entrenamiento.

Uno de los motivos por el que no se utilizó una ventana de entrada más grande fue la restricción de memoria del sistema operativo Win 3.1, ya que cada caracter de la ventana tiene asociadas 30 unidades, y cada una de ellas tiene un peso y un umbral por cada unidad de la primera capa oculta.

b) Palabras o segmentos de palabras no representados en el conjunto de entrenamiento

El criterio básico que se utilizó para seleccionar las palabras del conjunto de entrenamiento fue el siguiente: obtener un número equilibrado de palabras agudas, graves y esdrújulas de modo de presentar todos los casos distintos posibles que se deducen a partir de las reglas de acentuación. De acuerdo a dichas reglas, la acentuación de las palabras con acento implícito depende de la letra en que terminan, y en base a ese criterio se seleccionó un número mínimo

para cada tipo de palabra (aguda y grave). Otro criterio que se consideró fue que la ocurrencia de cada símbolo del alfabeto no estuviera muy bajo del promedio de ocurrencia de las letras en el conjunto de entrenamiento, para efectos de la traducción de texto a fonemas. Sin embargo, estos criterios no contemplan todas las distintas combinaciones en que pueden encontrarse los símbolos dentro de las palabras del español. La entrada de la red es un conjunto de 7 letras y, por lo tanto, el número de entradas posibles para la red, sin considerar el acento, es:

Número de casos posibles = (Número de símbolos de entrada) ^{7} = (28) ^{7} = 13.492.928.512

La mayor parte de estas combinaciones no existe en el español. No obstante, el número de combinaciones que si existe es muy grande y no es representado completamente por el conjunto de entrenamiento al considerar sólo los criterios mencionados.

Para mejorar el desempeño de las redes es necesario incorporar más palabras en el conjunto de entrenamiento con el objeto de cubrir combinaciones de palabras que no están presentes en el conjunto actual. Los conjuntos de validación y prueba también deben ser más grandes para medir en forma más representativa el desempeño de la red durante el entrenamiento y una vez finalizado este.

Cabe mencionar que puede haber habido sobreajuste en las redes entrenadas debido al tamaño del conjunto de validación (135 palabras; 1001 patrones).

3.3 Asistente de lectura

En la Tablas 34, 35 y 36 se presentan los resultados de las pruebas realizadas a 10 usuarios vendados para evaluar el funcionamiento del Asistente de Lectura. La Tabla 34 indica el grado de dificultad percibido por los usuarios al utilizar el Asistente de Lectura.

Tabla 34: Percepciones de los usuarios respecto de la dificultad en el uso del Asistente de Lectura.

GRADO DE DIFICULTAD		
DIFÍCIL	REGULAR	FÁCIL
3	5	2

La principal dificultad para los usuarios fue la memorización de las teclas y opciones para el control del Asistente de Lectura, y el procedimiento de recorrido de directorios para la búsqueda o almacenamiento de archivos. En la Tabla 35 se presenta el número de palabras reconocidas por los usuarios sobre un texto de 200 palabras pronunciado de corrido y otro texto de 200 palabras sueltas pronunciado una por una.

Tabla 35: Promedio de palabras reconocidas por los usuarios.

RECONOCIMIENTO DE PALABRAS				
	TEXTO CONTINUO DE 200 PALABRAS		200 PALABRAS SUELTAS	
	Primera vez	Tercera vez	Primera vez	Tercera vez
ESPAÑOL	153	182	127	148
CHILENO	180	197	143	170
INGLÉS	137	155	109	131

En cuanto al reconocimiento de palabras, en los tres casos hay un aumento debido al acostumbramiento por parte de los usuarios a la pronunciación del sintetizador de voz. El reconocimiento es mayor en el caso del texto continuo debido a que el contexto de las frases

entrega información para deducir algunas de las palabras que no se entienden bien, lo que no sucede con las palabras sueltas. Esta situación también ocurre en el habla humana. La síntesis en base a fonemas del español adaptados al chileno es la que da mejor resultado en el reconocimiento por parte de los usuarios. Los fonemas del español para letras como la “z”, la “c” y la “ll” y los fonemas tomados del inglés generan sonidos que no corresponden al habla de Chile, lo que se traduce en una menor inteligibilidad de la voz sintetizada. En la Tabla 36 se presentan las percepciones subjetivas de los usuarios respecto de la calidad de la voz del Asistente de Lectura, en cuanto a nitidez, gusto y naturalidad.

Tabla 36: Percepciones de los usuarios respecto de las características del sonido de voz del Asistente de Lectura.

	GUSTO		NATURALIDAD		NITIDEZ		
	AGRADABLE	DESAGRADABLE	NATURAL	ARTIFICIAL	BUENO	REGULAR	MALO
ESPAÑOL	0	10	0	10	0	5	5
CHILENO	0	10	0	10	0	6	4
INGLÉS	0	10	0	10	0	7	3

A los usuarios les pareció desagradable el sonido de las voces debido a que lo encontraron poco natural, es decir, se reconocía que no era una voz humana. En cuanto a la nitidez, la versión de español chileno con fonemas del inglés fue la mejor catalogada, aunque fue también la que presentó menor inteligibilidad. En particular, para la versión de español chileno, la pronunciación de los fonemas de las letras “s”, “ch”, “x” y “ñ” fueron mal catalogados porque no se ajustaban al sonido esperado.

4. CONCLUSIONES

El objetivo de este trabajo de título era desarrollar un sistema computarizado de conversión de texto a voz que permitiera obtener, a partir de un archivo de texto en español, una salida audible inteligible en español chileno. Para la generación de voz se utilizó un software que emite sonido a partir de una secuencia de símbolos y comandos; los símbolos representan fonemas y los comandos permiten realizar acentuación y pausas en la pronunciación del texto. Para la traducción de texto a fonemas con acentuación se realizaron desarrollos basados en dos métodos alternativos; reglas ortográficas y redes neuronales multi capas. Finalmente, se elaboró una aplicación llamada Asistente de Lectura como interfaz de apoyo a la lectura de no-videntes, que opera como un lector de texto en el cual están incorporados el software de generación de sonido y dos módulos de traducción de texto a fonemas con acentuación, uno basado en reglas y el otro en redes neuronales.

El mejor resultado en la traducción a fonemas y en la acentuación se obtuvo en el módulo basado en reglas ortográficas ya que el idioma español tiene características que permiten clasificar todos los casos, para los procesos de traducción a fonemas y de acentuación, en un conjunto finito de reglas por lo que no se producen errores en ninguna de estas dos tareas. El tiempo requerido para llevar a cabo la traducción a fonemas y la acentuación, mediante reglas, de las 1.491 palabras que componen el conjunto de entrenamiento de la red neuronal es de 0,11 segundos en un computador con procesador Pentium II con 288 MB de memoria RAM.

Para llevar a cabo la traducción a fonemas y la acentuación mediante redes neuronales, se desarrollaron varias alternativas, todas ellas basadas en un sistema llamado NETtalk[13], que realiza estas mismas tareas para el idioma inglés. Este es un sistema basado en redes neuronales de tipo perceptrón de múltiples capas, y tiene como entrada de la red una ventana que codifica 7 letras seguidas. La salida de la red corresponde al fonema asociado con la letra ubicada en la posición central de la ventana, y existe una unidad en la capa de salida que indica si la letra va acentuada o no. La traducción y acentuación del texto completo se obtiene desplazando la ventana a lo largo de todo el texto. Durante el entrenamiento, después de obtener las salidas de la red para cada patrón (ver figura 11), se realizó retropropagación cada vez que el error (diferencia entre el valor esperado y el valor obtenido) de al menos una unidad de la capa de salida superara el 10%. Se implementaron 18 arquitecturas de red: 6 para traducción de texto a fonemas, 6 para acentuación y 6 para traducción más acentuación. La capa de entrada es la igual para todas; la capa de salida es la misma para las 6 arquitecturas de cada objetivo, cambiando sólo el número de unidades de la capa oculta. Se utilizaron tres conjuntos de palabras: entrenamiento, validación y prueba. Después de 500 épocas por el conjunto de entrenamiento, se seleccionó para cada red el conjunto de pesos y umbrales de la época en que se obtuvo el menor número de errores en el conjunto de validación. Se consideró error a cada patrón o entrada en la que al menos una de las salidas tuviera una diferencia de más del 50% del valor esperado. Finalmente, para medir el desempeño de la red seleccionada con palabras independientes del entrenamiento y de la selección de la mejor época, se utilizó el número de errores de la red en el conjunto de prueba. Los tres conjuntos de palabras fueron construidos con los mismos dos criterios: cubrir en la forma más homogénea posible todos los casos de acentuación y de traducción a fonemas. Se seleccionó para las 6 arquitecturas de cada objetivo la que tuviera mejor desempeño y menor número de unidades en la capa oculta. Para cada arquitectura seleccionada, se realizaron 6 entrenamientos variando las condiciones iniciales, para medir su efecto en el desempeño de las redes. El número de unidades de la capa oculta de las arquitecturas con mejor desempeño fue el siguiente: 7 unidades para la red que sólo traduce y 30 unidades para la red que traduce y acentúa y para la que sólo acentúa.

En la traducción de texto a fonemas, tanto la red que traduce y acentúa como la que sólo traduce a fonemas no presentaron errores de clasificación en los conjuntos de entrenamiento, validación y prueba, es decir, las arquitecturas de red utilizadas permiten resolver el problema de la traducción de texto a fonemas.

Respecto de la acentuación, tanto la red que traduce y acentúa como la que sólo acentúa presentaron errores de clasificación en los conjuntos de entrenamiento, validación y prueba. En base a los resultados, no se pudo determinar que una arquitectura de red fuera mejor que la otra en la acentuación. La diferencia en el desempeño entre ambas redes fue menor que la variación en el desempeño cambiando las condiciones iniciales para una misma arquitectura.

Tiempos de procesamiento de las redes seleccionadas			
Etapa	**Traducción y Acentuación [s]**	**Traducción [s]**	**Acentuación [s]**
Inicialización de la red	0,2	0,2	0,2
Traducción al formato de entrada de red	0,2	0,1	0,2
Traducción a fonemas más acentuación	67,8	20,7	54,8
Total	68,2	21,0	55,2

El tiempo requerido por la red que traduce y acentúa para procesar las palabras del conjunto de entrenamiento fue 67,8 segundos. A este tiempo hay que sumar el utilizado en traducir las palabras al formato de entrada de la red, que fue 0,2 segundos y 0,2 segundos para la inicialización. Luego, el tiempo total del proceso mediante la red neuronal fue de 68,2 segundos, aproximadamente 620 veces el tiempo empleado por el sistema de reglas en el mismo procedimiento.

Se detectaron dos fuentes de error en la acentuación con las arquitecturas de red utilizadas: el tamaño de la ventana de entrada y los casos no contemplados en el conjunto de entrenamiento. Un problema inherente a la forma en que se entregan las entradas a la red es que la ventana de entrada no permite distinguir entre segmentos iguales del largo de la ventana, pero pertenecientes a palabras distintas. Este problema afecta cuando los segmentos tienen distinta acentuación entre sí. Por este motivo se produce una contradicción en el entrenamiento de la red, ya que para una misma entrada hay más de una acentuación posible. La segunda fuente de error proviene de las palabras no representadas por el conjunto de entrenamiento. Como la entrada de la red es un conjunto de 7 letras y el número de símbolos del alfabeto es 28 sin considerar el acento, el número de combinaciones de entradas posibles para la red es 13.492.928.512. Aunque la mayor parte de estas combinaciones no existe como palabra en el español, hay un gran número que no es representado al considerar sólo los criterios utilizados en la construcción del los conjuntos.

Para mejorar el desempeño en la acentuación con redes neuronales, se proponen las siguientes alternativas:

1. Aumentar los conjuntos de palabras (entrenamiento, validación y prueba) para cubrir los casos menos representados y para obtener medidas de desempeño más representativas.

2. Encontrar el máximo número de letras (MaxLetras) que puede haber antes de un acento en las palabras del español. Una vez determinado MaxLetras, usar una ventana de entrada de la

red con ese número de letras. Entrenar la red palabra por palabra ubicando, para cada palabra, la última letra en la posición final de la ventana, es decir, justificando las palabras a la derecha de la ventana. Si la palabra tiene menos letras que MaxLetras, llenar el resto con espacios; si tiene más letras, llenar la ventana de entrada con las últimas MaxLetras de la palabra. La salida de la red debe codificar en forma binaria la posición de la letra acentuada en la ventana de MaxLetras. Para ello, el número de unidades debe ser igual al exponente al que hay que elevar 2 para obtener la potencia de 2 más cercana a (igual o superior) a MaxLetras. En el caso particular de las palabras de los tres conjuntos utilizados en este trabajo (entrenamiento, validación y prueba) se encontró, después de acentuar los conjuntos mediante reglas, que MaxLetras era igual a 6. El mismo valor se obtuvo sobre las 80383 palabras de un lemario de uso libre de palabras del español [43], después de acentuarlas mediante reglas. Considerando 6 letras en la ventana de entrada, la capa de salida requiere 3 unidades ($2^3 = 8$).
Realizando el entrenamiento de esta manera, no se produce el problema de entradas de red iguales con distinta acentuación. Sería necesario realizar un estudio de las reglas para verificar si no puede existir una palabra con acento implícito en una letra anterior a la sexta antes de la última de la palabra o verificar empíricamente esta regla sobre un lemario aún más extenso.

3. Para una red que sólo realiza acentuación, se puede reducir el número símbolos de entrada a los siguientes, que son los relevantes para la acentuación mediante reglas: 1) vocal fuerte (a, e, o); 2) vocal débil (i, u); y 3) consonante. Para la última letra de las palabras es necesario agregar dos símbolos más: 4) consonantes n y s, ya que influyen directamente en la acentuación de las palabras agudas y graves y 5) espacio. Esto reduce considerablemente el número de unidades de la capa de entrada necesarias para codificar los símbolos de las palabras (de 29 unidades por símbolo se disminuye a 5). Si, adicionalmente, se utiliza el método de entrenamiento mencionado en el punto anterior, se tiene una red de 6 * 5 = 30 unidades en la capa de entrada y 3 en la capa de salida.
Esta disminución en el tamaño de la red debería provocar una disminución en el tiempo de procesamiento para generar la salida de la red.

4. Para minimizar el tamaño de la red que realiza sólo traducción a fonemas, el número de letras de la ventana de entrada se puede reducir al número de letras necesario para realizar la traducción mediante reglas, que es 3, con la letra a traducir ubicada en la posición central de la ventana. Con ello, la capa de entrada se reduce a 90 unidades (versus las 210 de las redes utilizadas en este trabajo)

5. Para realizar traducción más acentuación con redes neuronales, se pueden utilizar las redes propuestas en los puntos 2 y 3. El primer paso debe ser acentuar explícitamente las palabras con acento implícito mediante la red del punto 2 y luego, la salida de esta red, traducirla a fonemas con la red descrita en el punto 3. El orden debe ser primero acentuación y después traducción a fonemas, ya que en proceso de traducción a fonemas se pierde información necesaria para el proceso de acentuación.

6. Las redes obtenidas se pueden optimizar realizando podas, eliminando las conexiones con pesos muy bajos entre unidades, y reentrenando.

Sin embargo, aunque se mejore el desempeño de las redes, el número de operaciones requerido por las reglas para acentuar y traducir a fonemas una palabra va a ser siempre inferior al requerido por las redes neuronales, que sólo para traducir las palabras al formato de entrada de red realizan un número de operaciones similar al de la traducción a fonemas con reglas.

La conversión de texto a fonemas implementada en este trabajo de título no lleva a cabo la etapa de normalización del texto, en la que el texto de entrada es transformado en una

secuencia de frases compuestas por palabras y signos de puntuación. Esto significa, entre otras cosas, expandir las abreviaciones, traducir los números a palabras y analizar los caracteres no alfanuméricos. Esta etapa debe ser implementada para obtener un lector versátil que permita leer todo tipo de textos.

La aplicación Asistente de Lectura desarrollada como apoyo a la lectura de no-videntes realiza traducción de texto a fonemas más acentuación mediante reglas y mediante redes neuronales. La red utilizada para este efecto fue la que traduce y acentúa. Todo el control del programa se realiza mediante el teclado, enviando mensajes de voz al usuario para indicar sus acciones. El programa permite buscar y abrir archivos de texto, controlar la velocidad, volumen y el modo de la lectura, y seleccionar alguna de las nueve voces del sintetizador de voz. La aplicación fue diseñada para Windows 3.1, y puede ejecutarse en Windows 95 y 98 si se dispone del hardware adecuado (tarjeta de sonido AWE32, AWE64 o SBGold de Creative). Se realizaron pruebas con 10 usuarios vendados para evaluar el programa en cuanto a su funcionalidad y a la calidad de la voz sintetizada. Los encuestados consideraron que en promedio la dificultad de uso del programa era “regular”, en una escala de tres niveles subjetivos (difícil, regular, fácil). Del mismo modo, evaluaron como desagradable y artificial la síntesis con fonemas en inglés, español y español adaptado a Chile. La inteligibilidad mejor evaluada se obtuvo con los fonemas del español adaptado a Chile. Considerando esta evaluación, se concluye que aunque el sonido de voz generado por el software TextAssist permite un alto grado de inteligibilidad para el español chileno, no es satisfactorio en el resto de sus características.

Para mejorar la solución de lector computarizado para apoyo a la lectura de no-videntes se requiere de un software de síntesis de voz de mejor calidad de sonido en cuanto a la inteligibilidad, nitidez, naturalidad y que permita realizar control de la prosodia, es decir, el conjunto de pausas y entonaciones que dan naturalidad a la pronunciación de un texto, característica en la que TextAssist es muy limitado. Se requiere además que tenga un costo accesible y que funcione en los sistemas operativos más difundidos, como las distintas versiones de Windows y Linux. En cuanto al diseño de la interfaz de comandos de teclado para el control del lector de textos, sería necesario recopilar información sobre los estándares de teclas existentes para establecer una configuración de teclas por defecto y, por otro lado, obtener sugerencias por parte de los propios usuarios respecto de las funciones y combinaciones de teclas que ellos requieren. La interfaz diseñada debe tener la capacidad de permitir que los usuarios puedan reasignar las teclas y crear combinaciones de comandos propios. Sería útil incorporar una interfaz gráfica para que personas sin problemas de vista puedan dar soporte a los usuarios no videntes respecto de las funcionalidades del lector. Finalmente, el diseño de la interfaz debe ser modular para poder incorporar a futuro métodos alternativos de control tales como reconocimiento de voz, control remoto u otros medios que puedan servir para este propósito.
El Centro para Investigación de Tecnologías del Habla (CSTI) de la Universidad de Edimburgo, Escocia, ha desarrollado estándares, herramientas y documentación (Edinburgh Speech Tools) para el desarrollo de conversores de texto a voz basados en la concatenación de unidades acústicas (difonemas, mitades de fonemas, fonemas, trifonemas, sílabas), que son los sintetizadores que actualmente ofrecen mejor calidad de síntesis. Las ventajas de desarrollar un TTS (Text to Speech System) para el español chileno con estas herramientas son varias: rapidez para incorporar nuevos conocimientos en el área, nuevas personas en el desarrollo, disminuir errores, reducir el tiempo requerido para las distintas etapas de investigación y desarrollo y permitir la continuidad y permanente mejora del sistema. Se puede, por ejemplo, incorporar en forma modular las reglas de traducción de texto a fonemas y de acentuación implementadas en este trabajo de título. Las herramientas de CSTI están disponibles en forma gratuita y permiten desarrollar aplicaciones tanto en Windows como en Linux, con lo que se obtendría un sistema de buena calidad, accesible y de bajo costo o gratuito.

Para la construcción de voces en español chileno, se requiere de conocimientos específicos sobre fonética y lingüística. Los sistemas TTS en español tienen incorporado parte importante del conocimiento acumulado en estas áreas, pero hay características que son propias del español chileno que deben ser estudiadas y aplicadas en las etapas de generación de prosodia, en la definición de las unidades acústicas y en la creación del corpus de voz. Por ello, sería de gran utilidad contar con la colaboración de un fonoaudiólogo o un investigador del área. En cuanto a la grabación del corpus de voz, el o los locutores deben tener un control entrenado de su voz, además de características de voz amigable, nítida y natural.

Por último, para que el lector de textos desarrollado pueda ser utilizado con éxito por los usuarios no-videntes, es necesario que el sistema operativo o el ambiente de escritorio tenga incorporadas características que les permitan el acceso a los recursos del computador, al menos, para encender el computador, buscar y ejecutar la aplicación de lectura de textos. Esto actualmente es posible en algunos ambientes de escritorio de Linux.

Referencias

[1] Burger D., “Improved Access to Computers for the Visually Handicapped: New Prospects and Principles”, IEEE Transactions on Rehabilitation Engineering, vol.2, No.3, Sept. 1994, pp. 111-118.

[2] O’Malley M., “Text-to-Speech Conversion Technology”, Computer, Aug. 1990, pp. 17-23.

[3] Pérez C.A., Marín G., Holzmann C. and Valenzuela P., “A System for Non-Sequential Presentation of Vibrotactile Patterns for the Blind”, Abstracts of the World Congress on Medical Physics and Biomedical Engineering, Rio de Janeiro, Brazil, Aug. 1994, 21-26, pp. 864.

[4] Pérez C.A., Marín G., Holzmann C. and Valenzuela P., “Sistema para la Transferencia Táctil de Texto a No-Videntes”, X Congreso Chileno de Ingeniería Eléctrica, Valdivia, 1993, 22-26 Nov., pp. 49-53.

[5] Pérez C.A. and Marín G., “Handwritten Digit Recognition by a Neural Network with Slope Detection and Multiple Planes per Layer Architecture”, Proceedings of the 15th International Conference of the IEEE/EMBS, San Diego, CA, U.S.A, 1993, Oct. 28-31, pp. 275-276.

[6] El-Iman Y. and Banat K., “Text-to-Speech Conversion on a Personal Computer”, IEEE MICRO, Aug. 1990, pp. 62-74.

[7] Mniszewski S., “Teaching Computers to Talk Right”, IEEE Potentials, Feb. 1993, pp. 12-14.

[8] Hwang S. and Chen S., “Neural-network-based FO Text-to-speech synthesiser for Mandarin”, IEE Proc.- Vis. Image Signal Process., Vol. 141, N°6, Dec. 1994, pp. 384-390.

[9] Ritchie J., “Representación Digital y Síntesis de Voz”, memoria para optar al título de Ingeniero Civil Electricista, Depto. de Ingeniería Eléctrica, U. de Chile, 1986, 181p.

[10] Venturini R., “Diseño y Construcción de un Sintetizador de Voz en Español”, memoria para título de Ing. Ejec. en Electrónica, Depto. de Ingeniería Electrónica, U.T.F.S.M., 1990, 134p.

[11] Gibson D. L., Gilliott T. J. and Helliker L. A., “Texttalk: the British Telecom text-to-speech system”, Br. Telecom Technol J 6, No. 2, April 1988, pp. 157-170.

[12] Pisoni D. B., Nusbaum H. C. and Greene B. G., "Perception of synthetic speech generated by rule", Proc. IEEE 73, Digital Equipment Corporation, DTC-01-AA, 1985, pp. 1665.

[13] Sejnowski T. J. and Rosenberg C. R., “NETtalk: a parallel network that learns to read aloud”, The Johns Hopkins University Electrical Engineering and Computer Science Technical Report, 1986.

[14] Creative Labs, Inc., ”TextAssist user’s guide”, 1994.

[15] Creative Technology Ltd., “Creative TextAssist API Developer’s Guide”, Version 1.10, Sept. 23, 1994.

[16] Creative Technology Ltd., “Creative TextAssist API Developer’s Reference”, Version 1.10, Sept. 9, 1994.

[17] Lippmann R. P., “An introduction to computing with neural nets”, IEEE ASSP magazine, April 1987.

[18] Real Academia Española, Espasa Calpe S. A., “Gramática de la Lengua Española”, Madrid, 1970.

[19] Allen J., Hunnicutt M. S. and Klatt D., “From text to speech: the MITalk system”, MIT Press, Cambridge, Massachusetts. 1987.

[20] Johnson C. D., “Formal Aspects of Phonological Description”, Mouton, The Hague. 1972.

[21] Kaplan R. M. and Kay M., “Regular models of phonological rule systems”, Computational Linguistics, 1994, 20:3, 331-378.

[22] Karttunen L., Kaplan R. M. and Zaenen A., “Two-level morphology with composition”, Proceedings of the 14th International Conference on Computational Linguistics, Nantes, France, 1992, vol. 1, pp. 141-148.

[23] Pereira F., Rebecca C. N. and Wright N., “Finite state approximation of phrase structure grammars”, Proceedings of the 29th Annual Meeting of the Association for Computational Linguistics, Berkeley, California, USA. Association for Computational Linguistics, 1991, pp. 246-255.

[24] Coker C., Church K. and Liberman M., “Morphology and rhyming: Two powerful alternatives to letter-to-sound rules for speech synthesis”, Bailly, G. and Benoit, C., editors, Proceedings of the First ESCA Workshop on Speech Synthesis, Autrans, France. European Speech Communication Association, 1990, pp. 83-86.

[25] Golding A., “Pronouncing Names by a Combination of Case-Based and Rule-Based Reasoning”, PhD thesis, Stanford University, California, USA, 1991.

[26] Dedina M. and Nusbaum H., “PRONOUNCE: a program for pronunciation by analogy”, Computer Speech and Language, 1991, 5:55-64.

[27] Damper, R. I. and Eastmond J. F. G., “Pronunciation by analogy: Impact of implementational choices on performance”, Language and Speech 40(1), 1997, pp. 1-23.

[28] Daelemans W., van den Bosch A. and Weijters T., “IGTree: Using trees for compression and classification in lazy learning algorithms”, Artificial Intelligence Review 11, 1997, pp. 407-423.

[29] Roche E., “Two parsing algorithms by means of finite-state transducers”, Proceedings of the 15th International Conference on Computational Linguistics, Kyoto, Japan, 1994, pp. 431-435.

[30] Monaghan A., “Rhythm and stress in speech synthesis”, Computer Speech and Language, 1990, 4:71-78.

[31] Sproat R., “English noun-phrase accent prediction for text-to-speech”, Computer Speech and Language, 1994, 8:79-94.

[32] Church K., “Stress assignment in letter to sound rules for speech synthesis”, Proceedings of the 23rd Annual Meeting of the Association for Computational Linguistics, University of Chicago, Illinois, USA. Association for Computational Linguistics, 1985, pp. 246-253.

[33] Vitale T., “An algorithm for high accuracy name pronunciation by parametric speech synthesizer”, Computational Linguistics, 1991, 17:257-276.

[34] Marchand Y. and Damper R. I., “A Multi-Strategy Approach to Improving Pronunciation by Analogy”, Image, Speech and Intelligent Systems (ISIS) Research Group, Department of Electronics and Computer Science, University of Southampton, Southampton SO17 1BJ, UK, 2000.

[35] Beutnagel M., Conkie A., Schroeter J., Stylianou Y. and A. Syrdal., “THE AT&T NEXT-GEN TTS SYSTEM”, Proceedings of. Joint Meeting of ASA, EAA, AND DAGA, March. Berlin, Germany 1999.

[36] www.**bell-labs**.com/project/tts/tts-**overview**.html. Webmaster: webmaster@bell-labs.com, 21 Jun. 2005.

[37] Boula de Mareüil P. and Soulage B., “Input/Output Normalisation and Linguistic Analysis for a Multilingual Text-To-Speech Synthesis System”, paper 109, 4th ISCA Tutorial and Research Workshop (ITRW) on speech synthesis (SSW4), Perthshire, Scotland, 2001.

[38] Clark R., Richmond K. and King S., “Festival 2 - build your own general purpose unit selection speech synthesizer”, Proceedings of the 5th ISCA workshop on speech synthesis, Carnegie Mellon University, USA, 2004 pp. 173-178.

[39] Black A. and Lenzo K., “Flite: a small fast run-time synthesis engine”. Proceedings of the 4th ISCA Workshop on Speech Synthesis. Carnegie Mellon University, USA, 2001.

[40] Badino L., Barolo C., Quazza S., “Language Independent Phoneme Mapping For Foreign TTS”, Proceedings of the 5th ISCA Speech Synthesis Workshop – , Carnegie Mellon University, USA, 2004, pp. 217-218.

[41] Pave N., Gros J., Dobri S., Miheli F., “Homer II-man-machine interface to internet for blind and visually impaired people”, Computer Communications Volume 26, Issue 5, Laboratory of Artificial Perception, Systems and Cybernetics, Faculty of Electrical Engineering, University of Ljubljana, Slovenia, 20 March 2003, pp 438-443.

[42] NISO DTB. “ANSI/NISO Z39.86-2002. Specifications for the Digital Talking Book”, American National Standard Developed by the National Information Standards Organization, USA. 2002.

[43] http://lemarios.olea.org. Webmaster: Ismael Olea González, olea@hispalinux.es, 06 Ene. 2008.

ANEXO A : Avances en la conversión de texto a fonemas y en las aplicaciones de apoyo a la lectura de no-videntes

Son muchos los avances que ha habido en los últimos 10 años (1997-2005) en la conversión de texto a fonemas y en los sistemas de apoyo a la lectura de no-videntes. Se describen brevemente los métodos y aplicaciones computacionales más utilizados actualmente en ambos temas.

Avances en la conversión de texto a fonemas

En este trabajo, se presentaron dos alternativas para la conversión de texto a fonemas en español chileno, reglas y redes neuronales del tipo perceptrón de múltiples capas. Existen otras técnicas para la traducción de texto a fonemas que producen distintos resultados dependiendo del idioma. En algunas de ellas se considera como entrada una secuencia de símbolos en vez de uno solo y se obtiene como salida, a su vez, una secuencia de sonidos. Las técnicas más relevantes son las siguientes:

- Análisis morfológico: consiste en la descomposición de una palabra en sus partes morfológicas para obtener su pronunciación a partir de la pronunciación (si es que es conocida) de sus partes [19]. En el análisis morfológico se han utilizado con éxito autómatas de estado finito (FSA) para implementar las reglas de alternación morfológica [20, 21] y para representar diccionarios grandes [22, 23].
- Rotulado de partes del habla: es un proceso semejante al análisis morfológico, en donde se obtiene la pronunciación en base a partes de las palabras como raíces, prefijos y terminaciones, entre otras, y utilizando heurística para los casos ambiguos.
- Pronunciación por analogía: se obtiene la pronunciación de una palabra o de un segmento de una palabra por analogía con la pronunciación de otra palabra o segmento similar, cuya pronunciación sí es conocida. Este método está poco especificado como modelo teórico y para su implementación requiere de elecciones que sólo se resuelven mediante prueba y error [24, 25, 26, 27].
- Métodos de aprendizaje basados en memoria (memory-based learner): toman decisiones sobre la pronunciación de una palabra nueva en base a las decisiones similares tomadas anteriormente; para ello requieren almacenar el conjunto de entrenamiento completo. Una implementación para convertir grafemas a fonemas es el método IB1-IG [28]
- Transductores de estado finito (FST): se utilizan para representar reglas de dos niveles (como las reglas de traducción de texto a fonemas del español) e información de léxicos (reglas morfológicas) [29].
- Reglas semánticas y léxicas: se utilizan para detectar la acentuación; se basan en modelos estadísticos que se sustentan en corpus elaborados manualmente con información de acentuación [30, 31].
- N-gramas: permiten obtener la pronunciación de nombres propios, que frecuentemente son de distintas nacionalidades, a partir un modelo de secuencias de letras de varios idiomas [32, 33].

Los primeros cuatro métodos (análisis morfológico, rotulado de partes del habla, pronunciación por analogía y aprendizaje basado en memoria) son utilizados en idiomas que presentan mucha irregularidad entre la representación ortográfica y la fonética, como es el caso del inglés, idioma en el que el método de pronunciación por analogía es el que da mejor resultado para los casos de pronunciación desconocida [34]. Los nuevos sistemas de conversión a fonemas para el inglés utilizan combinaciones de las distintas técnicas.

Para el español, el método de reglas es el más efectivo de todos, tanto para la traducción de texto a fonemas como para la acentuación, debido a la regularidad que existe en la relación

entre sus representaciones ortográfica y fonética. Los sistemas de conversión de texto (TTS) más recientes utilizan transductores de estado finito para representar estas reglas.

Avances en la tecnología de apoyo a no videntes

Motores de conversión de texto a voz (TTS) y de síntesis de voz:

Un motor TTS (Text to Speech engine) es un conjunto de bibliotecas de funciones y bases de datos de voz que proveen servicios de conversión de texto a voz a las aplicaciones computacionales que los requieren. Un motor de síntesis de voz es similar, pero sólo realiza síntesis de voz. La arquitectura de estos sistemas puede variar desde motores y aplicaciones de escritorio ejecutándose en un mismo computador, a una arquitectura cliente/servidor, donde el cliente y el servidor pueden ejecutarse en distintos sistemas operativos y/o plataformas de hardware. Se destacan los siguientes motores TTS:

AT&T Natural Voices Text-To-Speech. [35]: Es uno de los motores TTS comerciales de mejor calidad tanto en la conversión a fonemas como en la síntesis de voz. Provee servicios a los desarrolladores de aplicaciones en múltiples plataformas de hardware y sistemas operativos para realizar conversión de texto a voz en varios idiomas: inglés estadounidense y británico, alemán, francés y español latinoamericano.

El texto pasa por tres etapas principales para ser sintetizado: normalización del texto, análisis lingüístico y síntesis de voz.

- Normalización: transforma el texto de entrada en una secuencia de frases compuestas por palabras y signos de puntuación. Para ello, se expanden las abreviaciones, se analizan los acrónimos y otros elementos que no son palabras.
- Análisis lingüístico: el texto normalizado se convierte en una representación lingüística que incluye los fonemas que deben ser producidos, su duración, la ubicación de los límites de las frases, y el contorno del tono o frecuencia para cada frase. No hay información disponible en forma pública respecto del método que utiliza para la conversión de texto a fonemas en español.
- Síntesis de voz: realiza síntesis por concatenación con selección de unidades. Utiliza mitades de fonemas (half phones) como unidad acústica, con lo que obtiene mayor naturalidad en el sonido. Para la minimizar el tiempo de proceso de selección de unidades, utiliza un caché de costos de concatenación para almacenar información de las unidades más utilizadas.

Bell Labs TTS. [36]: Es un producto comercial desarrollado en los laboratorios Bell Labs estructurado como una arquitectura de sistemas de conversión de texto a voz que realiza síntesis en varios idiomas: inglés, francés, español, italiano, alemán, ruso, rumano, chino y japonés. Las etapas necesarias para la síntesis son las mismas para todos los idiomas, salvo en inglés, y los datos específicos para cada lenguaje se almacenan en tablas externas que pueden leerse en tiempo de ejecución. El texto pasa por tres etapas principales para ser sintetizado:

- Análisis lingüístico: se analiza el texto para determinar los límites de las palabras, las componentes sintácticas, los límites de las frases y obtener la identidad de las unidades acústicas que se van a utilizar, el volumen, el tono, el contexto de las unidades adyacentes y la posición en la frase. Para ello usa transductores de estado finito ponderados (WFST).
- Modelamiento prosódico: toma la información proveniente del análisis lingüístico para asignar la duración y la entonación a cada segmento de voz.
- Síntesis de voz: lleva a cabo síntesis por concatenación de unidades de voz, principalmente difonemas, con selección de unidades. Una vez seleccionadas las unidades, se modifica su duración, tono y amplitud para ajustarse a la prosodia deseada.

ELAN TTS. [37]: Es un producto comercial que realiza traducción de texto a voz en ocho idiomas: francés, inglés, español, portugués brasileño, alemán, ruso, italiano y polaco. El texto

pasa por tres etapas principales para ser sintetizado: normalización del texto, análisis lingüístico y síntesis de voz.
- Normalización: en esta etapa se transforma el texto de entrada en una secuencia de frases compuestas por palabras y signos de puntuación. Esto se realiza mediante una serie de procesos basados en reglas, que difieren en cada idioma, donde se expanden las abreviaciones, se analizan los acrónimos, los números, direcciones de e-mail, los caracteres no alfanuméricos y otros elementos. Para ciertos idiomas es necesario, además, detectar dónde comienzan y terminan las palabras y las frases.
- Análisis lingüístico: el texto normalizado se transforma en una tabla de fonemas con la duración y el valor inicial y final del tono. Esto se realiza mediante cinco procesos: análisis morfológico, clasificación de partes del habla, análisis sintáctico (detectando el inicio y el final de las frases), conversión de grafema a fonema (detectando las sílabas y asignando acentuaciones) y, finalmente, generación de prosodia (parámetros de tono, duración e intensidad, además de las pausas). Para la conversión de grafemas a fonemas se adoptó un enfoque similar a un sistema experto, excepto para el inglés, en que se usó pronunciación por analogía. En la generación de prosodia en español se utilizó un diccionario de dependencias sintácticas y semánticas para la ubicación de las pausas.
- Síntesis de voz: se realiza a través de la selección de unidades acústicas, principalmente difonemas, la concatenación y finalmente la modificación prosódica de éstas.

Festival. [38]: Es, a la vez, un sistema TTS multilingüe (actualmente español e inglés británico y estadounidense) y una estructura general para la construcción y configuración de sistemas TTS tanto para investigación como para productos comerciales. Está disponible en forma gratuita para usos no militares y permite convertir texto a voz en múltiples plataformas de hardware y sistemas operativos. Requiere ser compilado en el computador en que se va a instalar.
Las etapas para la conversión de texto a fonemas son una serie de módulos configurables externamente que permiten independencia del idioma: lista de fonemas, lista de léxicos (formas generales de las palabras), detección de bordes de palabras, etiquetado de partes del habla, reglas para convertir letras a sonido (implementadas a través de transductores de estado finito ponderados o WFST).
La generación de prosodia también es un módulo configurable, en donde se asigna la duración y entonación de las unidades fonéticas. La síntesis de voz se realiza mediante la concatenación de difonemas.

Flite (Festival lite). [39]: Es un motor TTS gratuito, pequeño y rápido diseñado para servidores y sistemas incrustados (embedded systems), desarrollado en CMU (Carnegie Mellon University), y compatible con las voces del motor TTS Festival. Está programado en ANSI C y orientado a las arquitecturas ipaq (Linux/WinCe) y más pequeñas. Incluye una aplicación TTS que convierte texto en voz o archivos de audio, realizando síntesis por concatenación de difonemas. Proporciona una biblioteca de funciones para desarrollar aplicaciones.

Loquendo TTS. [40]: Es uno de los productos comerciales de mejor calidad para convertir texto a voz en 15 idiomas: italiano, inglés británico y estadounidense, alemán, francés, español (castellano, mejicano, argentino y chileno), catalán, portugués español, sueco, griego y chino (mandarín), desarrollado en Italia por Loquendo – Vocal Technology and Service. Realiza síntesis por concatenación con selección de unidades, utilizando difonemas y unidades más largas para obtener mayor naturalidad. En la conversión de texto a voz el texto pasa por las etapas de normalización, análisis lingüístico y síntesis de voz, pero no se ha publicado el procedimiento utilizado en cada una. Es el único que realiza síntesis en español chileno, pero tiene un alto costo.

Magnificadores de pantalla:
Son programas computacionales que permiten seleccionar y ampliar porciones de la pantalla del computador para que los usuarios que poseen ceguera parcial puedan acceder a la información. Generalmente se presentan como una funcionalidad de un sistema más grande. Un ejemplo de estos es MAGic 9.2 screen Magnification (2004)

Lectores de pantalla:
Son programas que permiten a los usuarios no-videntes o con ceguera parcial acceder al contenido de la pantalla del computador (principalmente al texto y, además, a algunos elementos gráficos como botones, barras de herramientas e íconos del escritorio). La información es entregada a través de un motor de conversión de texto a voz o mediante un despliegue en Braille. Los más populares son:
- JAWS (Job Access with Speech): es desde 1998 el lector de pantalla para Windows más usado por los discapacitados visuales españoles y latinoamericanos. Tiene salida de voz sintetizada y Braille.
- ZoomText (de la empresa estadounidense Ai Squared) es el magnificador y lector de texto más barato del mercado.
- Window Eyes screen reader.

Lectores de texto:
Son aplicaciones que utilizan un motor de conversión de texto a voz para leer el contenido de documentos de texto en formato digital (Word, PDF, HTML) en forma directa. Otras funcionalidades que pueden estar presentes son: acceso a Internet Explorer, posibilidad de uso de distintos motores TTS (cada uno con sus propias opciones de voces e idiomas), mantener una lista para colas de lectura, generar archivos de salida con la voz sintetizada en formato de audio (wav, wma, mp3), acceso al contenido del texto en un terminal de despliegue Braille. Los más conocidos en ambiente Windows son ReadPlease! y TextAloud (estos dos son los de mejor calidad y pueden utilizar, entre otros, el motor AT&T Natural Voices), Microsoft Reader, 2nd Speech Center, IBM Home Page Reader. Todos ellos tienen aplicaciones de texto a voz de propósito general y aplicaciones más específicas (orientadas, por ejemplo, a lectura del contenido de páginas web). Otro lector es Homer II [41], que es controlado por voz, desarrollado para los ciegos y discapacitados visuales para leer textos eslovenios en Linux y Windows.

Libros digitales parlantes (DTB o Digital Talking Book) y lectores de libros digitales parlantes:
Un libro digital parlante (Digital Talking Book o DTB [42]) es una colección de archivos en formato digital, que cumplen con el estándar DTB (ANSI/NISO Z39.86-2002), diseñados para presentar textos a usuarios con discapacidades físicas para la lectura (no-videntes o con discapacidad visual, discapacitados para el aprendizaje, impedidos físicamente y otros) a través de varios medios alternativos (grabaciones de voz humana, sintetizadores de voz, despliegues de Braille retráctil, magnificadores de pantalla). Un DTB permite un recorrido rápido y flexible del libro, marcar y destacar secciones o posiciones específicas, buscar palabras, controlar la pronunciación, configurar la presentación del contenido (notas al pie, número de página, etc.). El contenido de un DTB puede variar desde texto XML solamente hasta texto sincronizado con el audio correspondiente, o audio con poco o nada de texto. Para acceder al contenido de un DTB se requiere de un software especial de reproducción o de un dispositivo de hardware para reproducción de DTB. Los más conocidos son Victor Reader Soft de VisuAide (holandés, inglés estadounidense, francés, noruego), LP Player de Labyrinten, Dolphin Computer Access.

Editores de texto:
Son aplicaciones computacionales que utilizan un motor TTS para permitir a los usuarios no-videntes o con deficiencia visual generar documentos de texto, controlando todas las

funcionalidades del editor mediante el teclado. Se puede controlar la lectura del texto y almacenar en distintos formatos de audio. SayPad es un editor gratuito de este tipo.

Sistemas operativos y entornos de trabajo accesibles a no-videntes:
Las nuevas versiones de sistemas operativos y entornos de trabajo han integrado algunas de las tecnologías que permiten su utilización por parte de usuarios discapacitados.
- Sistema operativo Mac OS X de Apple Macintosh permite interactuar con el computador mediante sistemas de reconocimiento y síntesis de voz en inglés (PlainTalk speech synthesis technology). El reconocimiento permite el control de aplicaciones y elementos del escritorio. La síntesis de voz permite lectura de textos, escuchar mensajes de las aplicaciones y del sistema operativo.
- K Desktop Enviroment de Linux (KDE): es un ambiente de escritorio gratuito de Linux orientado a que los usuarios discapacitados tengan acceso al escritorio y la tecnología subyacente en la forma más eficiente posible proporcionando un acceso fácil a todos los elementos de los programas a través del teclado. Esto es logrado a través de APIs, que permiten acceder a todas las interfaces gráficas de una aplicación mediante el uso de las tecnologías existentes de apoyo a discapacitados, tales como lectores de pantalla, controladores de dispositivos Braille, teclados de pantalla y otras. Para la síntesis de voz, requiere la instalación de un motor TTS (Festival, MBROLA, txt2fho, freeTTS, Flite).

Estándares de lenguajes de marcas (markup languages) para voz:
Son lenguajes que permiten a las aplicaciones clientes de un motor TTS incluir marcas en el texto de entrada que cambian la manera en que se va a sintetizar. Estas marcas permiten, por ejemplo, controlar la pronunciación específica de una frase o palabra indicando cada fonema; controlar la síntesis de números (fracciones, decimales, etc.), sintetizar texto con fechas, direcciones, o números de teléfono; cambiar la voz, el volumen y la velocidad; cambiar la base de datos de voz para seleccionar una pronunciación de una frase o palabra. Los lenguajes de marcas para voz más difundidos son Java Speech Markup Language (JSML), Speech Synthesis Markup Language (SSML, componente del estándar Voice XML), Microsoft SAPI 4.0 y Microsoft SAPI 5.1.

Dispositivos de Hardware

Dispositivos en lenguaje Braille: existen terminales de despliegue Braille de una o varias filas, para computadores personales, teléfonos fijos y móviles (entrada y salida). La empresa más conocida es la holandesa ALVA Braille Products.

Lectores de libros digitales parlantes (DTB): existen dispositivos lectores para el formato DAISY (Victor Reader, de VisuAide y Plextalk, de Plextor).

Áreas de investigación en la conversión de texto a voz en el 2005

Las siguientes son actualmente las principales áreas de investigación en la generación de voz sintetizada:

Desarrollo de mejores y más completas prueba de calidad de los sistemas TTS que permitan evaluar y comparar la mayor cantidad posible de características de cada etapa de la conversión de texto a voz. Se destacan en éste ámbito las pruebas con oyentes (listener tests).

Mejoras en el control y variedad de características prosódicas de los sistemas TTS. Entre los estilos prosódicos que se han incorporado a los nuevos sistemas están: tipos de frases o

sentencias (afirmaciones, preguntas, exclamaciones y otras), modos de lectura (continua, deletreo, diálogos y otras), emociones (alegría, ira, disgusto, miedo, sorpresa, tristeza). Los sistemas disponen de varias bases de datos de voz con cada estilo. Para ello se han estudiado las características prosódicas que permiten estimar curvas de entonación y se han desarrollado herramientas de obtención automática de características prosódicas a partir de bases de datos extraídas de voz natural.

TTS políglotas: tienen como objetivo realizar síntesis de voz adecuada para textos que tienen mezcla de varios idiomas.

Existen otras áreas de investigación en donde la generación de voz es sólo una parte del sistema, entre las que se destacan principalmente los sistemas de diálogo hablado (Spoken Dialog Systems), en los cuales el sistema interactúa con el usuario y genera respuestas de voz de acuerdo a esta interacción, utilizando en este proceso tanto reconocimiento como síntesis de voz.

ANEXO B : Redes neuronales de tipo perceptrón de múltiples capas y algoritmo de retropropagación

Las redes neuronales del tipo perceptrón de múltiples capas [13], [17] consisten en un gran número de unidades de cálculo que se agrupan jerárquicamente en capas. Una capa de entrada posee unidades que reciben valores de entrada de red, y una capa de salida tiene unidades que transmiten valores o resultados. Cada unidad de la red recibe un conjunto de entradas de valor continuo y genera una salida a partir de la suma de esas entradas, mediante una transformación no lineal. Las unidades de una capa están unidas a las de la siguiente a través de pesos, que pueden tener valores reales positivos o negativos, representando una influencia excitativa o inhibitoria de la primera unidad sobre la segunda. Cada unidad tiene, además, un umbral, que es sustraído de la suma de las entradas a dicha unidad. Este umbral puede considerarse como el peso de una unidad que tiene valor fijo 1. De este modo, la notación y el algoritmo de aprendizaje pueden aplicarse tanto para los pesos como para los umbrales. Para obtener la salida de la i-*ésima* unidad de la capa n+1, es necesario calcular la suma ponderada de sus entradas:

$$E_i = \sum_j w_{ij} p_j \qquad \textbf{(1)}$$

donde p_j es la salida de la unidad j-*ésima* de la capa n y w_{ij} es el peso desde la unidad j-*ésima* de la capa n a la unidad i-*ésima* de la capa n+1. Hecha la suma, se aplica la función de transferencia sigmoide, con lo que la función de activación o salida de la unidad i-ésima queda entonces como:

$$p_i = \mathrm{P}(E_i) = \frac{1}{1 + \mathrm{e}^{-E_i}} \qquad \textbf{(2)}$$

Las unidades de la capa de entrada, a diferencia de las demás, reciben valores de entrada que no provienen de una capa anterior. Las unidades de la capa de salida, por su parte, no tienen pesos pues sus salidas no son entrada de otras unidades. El resto de las capas, que se encuentran entre la de entrada y la de salida, se denominan capas ocultas, y pueden ser una o varias. La información se propaga desde la capa de entrada hacia las capas siguientes, hasta generar las salidas de la última capa.

La red es capaz de “adquirir conocimiento” y para ello es necesario realizar un entrenamiento, en el cual se le entrega a la red un conjunto patrones de entrada y las salidas esperadas para cada patrón.

El algoritmo de retropropagación del error [17] es uno de los métodos de aprendizaje más utilizados en las redes neuronales. En este algoritmo, los pesos son ajustados incrementalmente durante el aprendizaje en base a la diferencia entre los valores esperados y los obtenidos en las unidades de la capa de salida. Para cada patrón de entrenamiento, este error es propagado hacia atrás (retropropagado) desde la capa de salida hacia la capa de entrada. Cada peso de la red es ajustado para minimizar su contribución al error cuadrático medio entre la salida esperada y la obtenida. La ecuación de ajuste de cada peso es la siguiente:

$$w_{ij}^{(n)}(t+1) = \alpha\, w_{ij}^{(n)}(t) + (1-\alpha)\varepsilon\delta_i^{(n+1)} p_j^{(n)} \qquad \textbf{(3)}$$

donde $w_{ij}^{(n)}$ es el peso desde la unidad j-*ésima* en la capa n hacia la unidad i-*ésima* en la capa n+1, el parámetro α suaviza el gradiente y ε controla la tasa de aprendizaje.

El error de la penúltima capa, $\delta_i^{(N-1)}$, es calculado a partir de la capa de salida N:

$$\delta i^{(N)} = (p_i^{*} - p_i^{(N)})P'(E_i^{(N)}) \qquad \text{(Para la capa N)} \qquad \textbf{(4)}$$

y es retropropagado recursivamente hacia las capas anteriores:

$$\delta i^{(n)} = \sum_j \delta_j^{(n+1)} w_{ij}^{(n)} P'(E_i^{(n)}) \qquad \text{(Para las capas anteriores a la N)} \qquad \textbf{(5)}$$

donde P'(E) es la primera derivada de P(E), p_i^{*} es la salida esperada para la i-*ésima* unidad en la capa de salida, y $p_i^{(n)}$ es el valor obtenido por la red en la i-*ésima* unidad, en la capa n. Generalmente, el error es retropropagado en la red cuando la diferencia entre el valor obtenido y esperado supera un valor arbitrariamente fijado en cada problema.

Para incrementar la tasa de aprendizaje sin caer en oscilaciones se utiliza un término denominado *moméntum* que corresponde al porcentaje de la variación de los pesos y umbrales de la época anterior que se considera para la época actual. De este modo, la variación para la época actual queda como la variación actual ponderada por la tasa de aprendizaje ε más la variación anterior ponderada por el factor moméntum, tanto para los pesos como para los umbrales.

Los pesos de la red deben ser inicializados con valores no nulos, pequeños y diferentes entre sí, puesto que el error se retropropaga a través de dichos pesos en proporción a los valores de ellos. Si fueran nulos, el sistema no tendría variación y si fueran idénticos, la red no sería capaz de aprender soluciones que requieren pesos diferentes. Los valores deben ser pequeños para no causar un desequilibrio que impida que la convergencia de la red. Para ello, se utiliza un parámetro (Random Gate) que establece la cota máxima para los valores iniciales, que son asignados en forma aleatoria para romper la simetría.

Existen dos modos de entrenamiento: por patrón y por época. En el primer caso, se calcula el error en las salidas después de cada patrón, y si el error supera el porcentaje máximo establecido, se realiza la retropropagación antes de procesar el siguiente patrón. Si el entrenamiento es por época, se acumula el error de cada patrón hasta el fin del conjunto de entrenamiento y si el error supera la cota máxima establecida, se realiza la retropropagación una vez finalizada la época.

ANEXO C : TextAssist y TAAPI

1. TextAssist

TextAssist es un conjunto de aplicaciones de síntesis de voz desarrolladas para Windows por CREATIVE LABS para los modelos de tarjeta de sonido Sound Blaster 16 ASP hasta el modelo GOLD 64. De todas ellas, la más relevante es Texto'LE.

Texto'LE es una aplicación que realiza síntesis de voz a partir de un texto. El texto puede ser editado directamente en una ventana de edición de la misma aplicación, o puede ser obtenido a partir de un archivo.

Los procesos realizados sobre el texto para generar la voz que lo pronuncia son transparentes al usuario, es decir, no hay acceso al conjunto de fonemas, acentuación, comandos de pausa y de prosodia previos a la pronunciación.

El usuario puede ejecutar los siguientes comandos sobre el texto (ya sea mediante el "mouse" o combinaciones de teclas):

- Comenzar la pronunciación del texto
- Dejar en pausa la pronunciación
- Continuar con la pronunciación
- Detener la pronunciación

El resto de los parámetros son modificables mediante comandos de menú de barra y del "mouse".

Texto'LE posee nueve voces diferentes para realizar la lectura. Sobre estas nueve voces se pueden efectuar las siguientes modificaciones:

- Aumentar/disminuir el volumen
- Aumentar/disminuir la velocidad de pronunciación (N° de palabras por minuto).
- Modificar el tono de la voz.

Existen dos modalidades de pronunciación del texto:

- Oración: el texto es leído en forma continua de comienzo a fin
- Palabra: la lectura se detiene después de cada palabra.

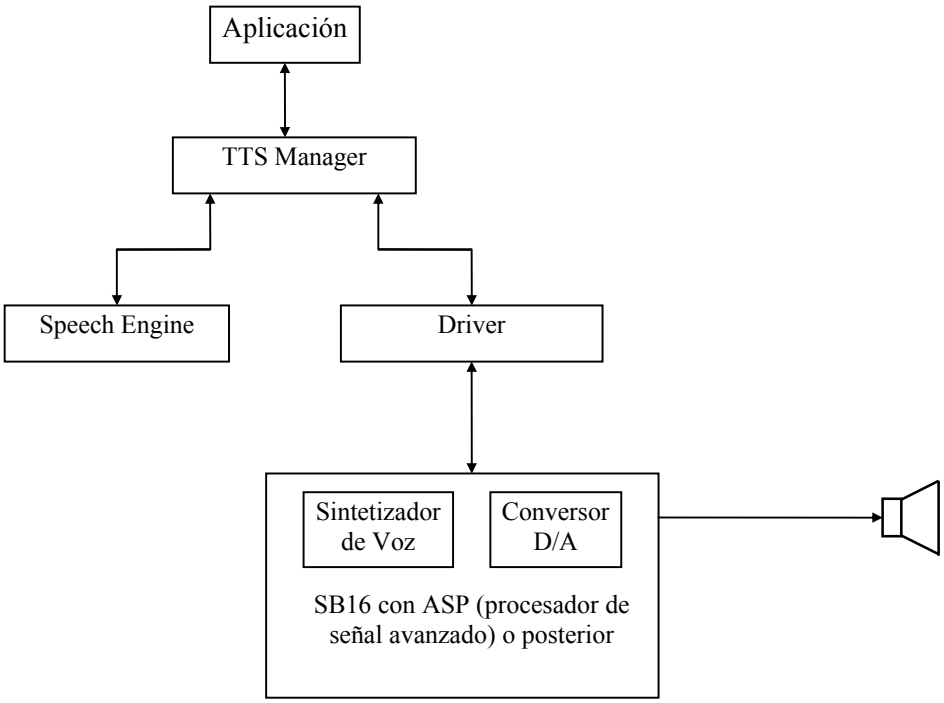
Otros parámetros modificables son el largo de las pausas asociadas a los diferentes signos de puntuación. En esta aplicación, se clasifican en dos grupos, de acuerdo al largo de la pausa: punto y coma.

Existen, además, una serie de comandos para crear texto cantado, que controlan la frecuencia y la duración de cada fonema. Mediante estos comandos, es posible realizar un control de la prosodia del texto, pero en un grado limitado.

2. TAAPI

Es un paquete de desarrollo elaborado por CREATIVE LABS para crear aplicaciones que requieren un control especial para síntesis de voz.

Contiene una biblioteca de enlace dinámico para Windows (ctsframe.dll) y soporte para programación en C, Visual Basic y Pascal.



La aplicación hace uso de los servicios de conversión de texto a voz a través del TTS Manager (Text-to-Speech). Cuando una aplicación entrega un texto al TTS Manager para ser pronunciado, este lo copia en un búfer interno y devuelve el control a la aplicación, permitiendo la reproducción de voz asincrónica. El resto del procesamiento de conversión de texto a voz es realizado en forma independiente. El TTS manager detecta los signos de puntuación para dividir el texto en segmentos. Cada segmento es entregado, uno por uno, al motor de voz. Esta usa un diccionario interno, un diccionario de excepciones, y reglas de pronunciación de lenguaje para convertir cada segmento de texto en fonemas con información de duración y frecuencia. Finalmente, la máquina de voz convierte los fonemas en parámetros de control del sintetizador, que contienen frecuencia de formantes, amplitud de la fuente de sonido y otra información utilizada por el Sintetizador de Voz para la reproducción de la síntesis de voz. Los parámetros de control del sintetizador son entonces devueltos al TTS Manager y entregados al Sintetizador de Voz a través del controlador del dispositivo. El Sintetizador de Voz, actuando en la tarjeta de sonido (SB con ASP), convierte los parámetros de control de sintetización en paquetes de forma de onda que son enviados al conversor D/A cada 6.4 milisegundos. El conversor D/A convierte los paquetes de forma de onda en sonido analógico que es enviado a los parlantes.

La interfaz TAAPI (TextAssist Aplication Program Interface) es un conjunto de 39 funciones en C. Parte de estas funciones son para procedimientos de inicialización al comienzo de la aplicación, y de limpieza antes del cierre de la misma. Hay un conjunto de funciones que permiten la reproducción de archivos de audio con formato *wav* y funciones específicas para el control de la síntesis de voz.

Dentro de este último grupo, las funciones utilizadas por la aplicación Asistente de Lectura para no/videntes son:

- ctsGetVoice

Retorna el número que identifica a la voz que está actualmente seleccionada.

- ctsPauseSpeech

Detiene momentáneamente la pronunciación del texto que se esté leyendo (si lo hay), hasta que se ejecute la función ctsResumeSpeech o ctsStopSpeech.

- ctsPlaySpeech

Comienza la pronunciación del texto.

- ctsResumeSpeech

Reanuda la pronunciación del texto desde donde quedó suspendida, si se había efectuado una pausa con ctsPauseSpeech.

- ctsSelectVoice

Selecciona una de las nueve voces que provee TextAssist.

- ctsSetSpeechParams

Fija uno o varios de los siguientes parámetros en el valor especificado en la misma función: pausa de la coma, pausa del punto, velocidad de pronunciación, volumen de pronunciación, frecuencia de la voz.

- ctsStopSpeech

Detiene la pronunciación del texto.

ANEXO D : Listas de palabras de los conjuntos de entrenamiento, validación y de prueba

1. CONJUNTO DE ENTRENAMIENTO

A continuación se presenta el listado de 1491 palabras del conjunto de entrenamiento, separado en palabras con acento implícito y explícito, y subdividido en palabras monosílabas, agudas, graves y esdrújulas. Para representar la vocal acentuada en las palabras con acento implícito, se utiliza el tilde invertido (`).

Palabras con acento implícito

Monosílabos

là	trèn	ñòr	lèy
yà	quièn	pòr	còy
clùb	buìn	gùr	dòy
grùb	fìn	sùr	hòy
dè	güìn	vàs	sòy
fuè	ruìn	tràs	vòy
gòlf	sìn	piès	hùy
wòlf	clòn	puès	mùy
fuì	còn	grìs	gàs
sì	zòn	luìs	pàz
bòj	gùn	lòs	juèz
gròj	ùn	tòs	nuèz
griàl	diò	gù	clìz
gàl	yò	ñù	ruìz
èl	fàr	frày	gòz
pièl	jàr	guày	vòz
flàn	güèr	hày	crùz
gràn	vèr	buèy	lùz
juàn	gruìr	dèy	
kàn	huìr	grèy	
bièn	ìr	juèy	

Agudas

actividàd	gucdòd	kuañèg	wognòj
antigüedàd	guectòd	kukìg	gambùj
capacidàd	güefròd	ñuquìg	guecdùj
dignidàd	jucnòd	weiyìng	wexùj
esterilidàd	kuyòd	gocdòg	axiàl
eternidàd	wilòrd	kingkòng	cañaveràl
exigüidàd	alamùd	wekòg	casuàl
habilidàd	algùnd	walrùg	cigüeñàl
incomodidàd	acdiñùd	wayùng	chaguàl
majestàd	cecnitùd	wikùg	chaparràl
cedèd	decrepitùd	balàj	fractàl
hacèd	exactitùd	erràj	generàl
koccèd	juventùd	maniblàj	glaciàl
parèd	magnitùd	valàj	herreñàl
vencèd	pulcritùd	gacnèj	iguàl
weñèd	virtùd	relèj	inmoràl
adalìd	walùd	woccièj	irreàl
ardìd	awàng	fuquìj	lineàl
gocnìd	jauvàg	ñuvìj	oxipitàl
madrìd	zigzàg	wakìj	rituàl
servìd	yingsèng	kuvòj	temporàl
suplìd	kevèg	relòj	veinteñàl

infièl
israèl
joèl
mantèl
pastèl
añìl
carrìl
chigüìl
doñeguìl
guayaquìl
mandrìl
aerosòl
alcohòl
autogòl
fajòl
ñielòl
abedùl
azùl
gandùl
garzùl
huemùl
ñikàm
kicnàm
zuguinàm
cruvèm
gucnoñèm
harèm
cluñìm
juñoguìm
walrìm
ñignòm
quixòm
wigòm
guecnùm
quesrùm
wolrùm
abjuràr
aguardàr
ahondàr
ahorcàr
añoràr
arrollàr
buceàr
callàr
collàr
coordinàr
coxqueàr
chapuceàr
dibujàr
desraizàr
desrielàr
desroñàr
enrayàr
exasperàr
extirpàr
fluctuàr
guardàr
hojeàr
hurgueteàr
injuriàr
loàr
malrotàr
obsequiàr
renunciàr
sonrojàr
vulgàr
alfilèr
caèr
comèr
creèr
embellecèr
extraèr
hacèr
leèr
llaèr
mordèr
mujèr
proveèr
raèr
roèr
traèr
yuxtaponèr
argüìr
construìr
corregìr
crujìr
disminuìr
exhibìr
exigìr
prohibìr
reducìr
rugìr
alrededòr
aprehensòr
erròr
expugnadòr
malhechòr
tractòr
vigòr
astùr
azùr
lientùr
mercosùr
segùr
aninàt
parmalàt
wogniàt
bufèt
corsèt
chalèt
devènt
jafèt
cenìt
edìt
efrìt
judìt
fagòt
merlòt
ramòt
robòt
algùnt
azimùt
calicùt
mamùt
gambàx
guagàx
relàx
kilroñèx
ucditèx
weccièx
ñuglìx
usreñìx
wogüìx
gambòx
güeclòx
wognòx
ñoctùx
güavùx
welrùx
achachày
ajajày
ayayày
candrày
carày
curibày
curupày
guirigày
nanày
norày
ñandubày
pacày
paraguày
uruguày
verdegày
yatày
camagüèy
canèy
carèy
curamagüèy
curujèy
jersèy
mamèy
pejerrèy
virrèy
yarèy
convòy
choròy
elòy
estòy
godòy
morrocòy
rentòy
tipòy
tongòy
cuicùy
güignùy
guecdùy
mangachapùy
wanrùy
capàz
falàz
mordàz
procàz
rapàz
acidèz
estupidèz
flaccidèz
soèz
timidèz
actrìz
barnìz
cervìz
narìz
perdìz
arròz
atròz
badajòz
feròz
velòz
arcabùz
avestrùz
calalùz
cuzcùz
micifùz
werrallùz

Graves

aguànta
aliànza
almohàda
aràña
àscua
atarràya
cañàda
càusa
diafràgma
dràma
excusàña
fragància
guadàña
güèctia
güirrècnia
guitàrra
hàcha
ignàvia
iguàna
jiràfa
llàrrua
llàuka
màgma
miàja
nicaràgua
pizàrra
psiquiàtra
tàcna
urràulla
wànda
yàya
agüèla
aldèa
aldehuèla
audiofrecuència
ceguèra
corregüèla
creència
dèuda
ecnotècnia

encèlla
exagèra
faèna
güèña
habichuèla
higüèla
higuèra
jimèna
lengüèta
noguèra
ñècla
oxigèna
quèja
rèina
sanguijuèla
vehemència
vejigüèla
vergüènza
vergüèña
zarigüèya
agìta
agüìta
anguìla
asraelìta
axìla
cañìta
estricnìna
excogìta
guagüìta
guìlla
güìmba
güìra
horquìlla
jarrìlla
lengüìta
macagüìta
maguìla
mojarrìlla
neguìlla
paragüìta
ruìna
tagüìta
traguìlla
vagìna
vajìlla
vìsta
yanìlla
yuquìlla
ahòra
alcachòfa
bellòta
bòa
caòba
cuòta
gaviòta
hòla
hònra
ingeniòsa
lòa
ñuñòa
òstra
parròquia
vòdka
zozòbra
alrùta
anacrùsa
basùra
coyùnda
creatùra
cùna
challùlla
disruptùra
diùca
ecdilrùra
enrejadùra
escritùra
excùsa
excrùta
exhùma
glùma
hallùlla
jùngla
kalrùña
kicdùcia
lùna
llùvia
marrùlla
ricdùra
signatùra
trùcha
vectùra
zùlma
achàque
bañàrse
camuflàje
desàgüe
desayunàrse
enrollàste
exàngüe
fiàmbre
gigànte
guànte
menguànte
ràcne
sàke
sàstre
wàgüe
wirràgüe
colrègue
coñète
corriènte
chèque
diènte
enclènque
enriquècе
èñe
exigènte
exponènte
higiène
incoherènte
llègue
nicaragüènse
orfèbre
pesèbre
pretendiènte
turgènte
berrìnche
bilìngüe
buìtre
curiquìngue
chìnche
chiquigüìte
chirìgüe
exequìble
inexequìble
jengìbre
kìrie
lìngue
ñìrre
yungìste
albaricòque
azògue
bròche
gaòsre
gasròcte
juestòke
pònche
ròble
woctògue
bùgle
deglùte
enchùfe
gactùche
grùñe
jùzgue
muchedùmbre
rùge
tiùque
càki
càqui
càsi
màqui
misràji
ñàchi
bèti
rèmi
yèni
yèti
fetuchìni
ìnri
kìli
paganìni
tortelìni
wìrri
ambrosòli
ñòqui
paròdi
raviòli
yògui
cùrsi
cùti
escùti
mùsri
mùti
dàrwin
clàxon
exàmen
frànklin
imàgen
màrgen
chorrèan
gèrmen
güèrren
lèen
nèwton
provèen
aniquìlen
ubìquen
ulrìksen
vìrgen
wìlson
arròjen
chòquen
empòllan
gestiònen
sòplan
arrùguen
ayùdan
cardùmen
cerùmen
enchùfan
antàño
aplàudo
arcàico
cañonàzo
chadiàno
chafarràño
chàncho
enredàdo
enrejàdo
ermitàño
extràño
giànzo
guàpo
guàyo
guijàrro
huàso
humillàdo
jàrro
kantiàno
kilogràmo
kognàdo
lenguàdo
poblàño
ràudo
recnidàrio
subsidiàrio
wellàuko
wolfràmio
worràiko
zafarràncho
zodiàco
acècho
agüèro
almizclèño
arroyuèlo
becèrro
carrèño
cencèrro
chorrèo
cohècho
corrèo
fèudo
gargüèro
güèlfo
güikèo
halagüèño
higüèro
huèvo
igüèdo

lièNzo
llollèo
llukèo
malaguèño
manriquèño
nèutro
perpètuo
pizarrèño
plèito
psèudo
quièto
recibimièNto
sangüèño
sonrisuèño
taxquèño
tacnèño
varguèño
wolrugèrio
yèrro
yuguèro
ambìguo
ceñìglo
cognitìvo
coquìllo
corrìllo
chigüìro
dìcho
egìpcio
equìno
exclusìvo
exhaustìvo
guìncho
güìro
guìso
ìstmo
junquìllo
kilocìclo
mediquìllo
morrìllo
occìduo
paquetìllo
pingüìno
pocìllo
sìsmo
suìzo
vikìngo
vestìglo
visìllo
wìldo
zorrìllo
acuòso
bizcòcho
cachòrro
caprichòso
còito
contagiòso
chinchòrro
desahògo
equinòccio
fervoròso
giroscòpio
gòlfo
ignominiòso
jactanciòso
kimòno
kiòsco
licnòbio
llòiko
morròño
ñòño
òdio
òro
plòmo
provechòso
religiòso
sinuòso
villòrrio
virtuòso
acuedùcto
añuñùco
cochayùyo
chùño
chùrro
clarùño
involùcro
jùnio
negrùzco
ñùrdo
occipùcio
orgùllo
sùsto
trùco
warrùllo
welrùsco
yeyùno
yùyo
àgnus
ambigüedàdes
cuñàdos
galàxias
garràfas
làicos
llàwas
màyas
milràyas
nostradàmus
oàsis
taràdos
alinèas
aquèllos
bayonètas
crèes
enagüètas
gregüèscos
guèrras
kèrmes
llurrèes
patèas
utrèras
wollèas
zaragüèyes
zèus
añìcos
corrìges
chillìdos
enagüìllas
fuìmos
gemìdos
güìrris
ìris
ecdìsis
wìgües
wìndows
alcohòles
cirròsis
convecciònes
estòicas
fricciònas
krònos
lòicas
reflexiònes
ròes
wirròikas
avestrùces
dùdas
frùtos
nùpcias
pantùflas
viùdos
zùmos
clàxu
llàllu
ñiàlru
wàctu
acroèpu
güècru
guèsru
güìctu
guìnru
trìbu
gañòctu
güillòrru
gocciòñu
wesròGnu
ñùglu
kùclu
wecnùgu

Palabras con acento explícito

Monosílabos

qué
mí
tú
quién
dió
fué
fuí
tí
pués
cuál

Esdrújulas

aerodinámica
cátedra
gráfica
máquina
táctica
espontáneamente
llámame
pirámide
quemándose
básquetbol
aerostático
ámbito
arácnido
árbitro
áspero
balsámico
catálogo
cuákero
dogmático
drástico
ecdático
ecdáceo
eclesiástico
enroñándolo
güisácdiko
hábito
icnográfico
lácneo
lánguido
obstáculo
pálido
plácido
quirográfico
rácdico
tecnográfico
xilográfico
xilrático
aclarándomelos
álamos
análisis

enfriándoselas
anécdota
célula
génova
molécula
sémola
célebre
célibe
gésrique
hélice
régimen
sinécdoque
antiaéreo
atmosférico
céntimo
confundiéndotelo
degüéllalo
diurético
ético
excéntrico
éxito
éxodo
gélido
gémino
género
glicérido
hermético
kaquéctico
magnético
géiser

énfasis
exégesis
éxtasis
géminis
génesis
gérmenes
huéspedes
miércoles
paréntesis
déficit
gloxínea
lingüística
orquídea
políglota
química
víbora
egílope
índice
quísroke
currículum
acrílico
barítono
bolígrafo
centrífugo
cíclico
cítara
cítrico
gíralo
ignívomo
lícito

líquido
mamífero
místico
pícnico
ridículo
tímpano
síntesis
subíamos
vírgenes
góndola
mayólika
ópera
póliza
apócope
acrónimo
alcohólico
anecdótico
arquitectónico
corpóreo
dextrógiro
diagnóstico
electrocardiógrafo
exógeno
fenómeno
gastrónomo
glaciólogo
helicóptero
homólogo
inkócdito
irónico

kinesiólogo
lexicógrafo
lóbrego
micnóbico
óptico
picnómetro
psicológico
sólido
vómito
zoológico
hipótesis
mayúscula
música
pústula
ayúntele
distribúyale
fúnebre
lúgubre
múltiple
búfalo
chúpalo
empúñelo
excúlpalo
glúteo
húmedo
icdúcido
júbilo
núcleo
siútico
útero

Agudas

acá
allá
está
guecniká
güicnorá
jacarandá
maracuyá
requecdá
wogondá
wonrurá
creíd
cloíd
glaíd
juñoíd
ataúd
greúd
jugoúd
laúd
acné
aguijoneé
ahorqué
alrecé
anuncié
arañé
arranqué
arresté
arrugué
claqué
delasolré
desreglé
enroñé
escrutiñé

exicniqué
expliqué
expurgué
forré
glasé
guiñé
hinché
intoxiqué
kalrigué
ñangué
rasguñé
rogué
santigüé
vengué
yaqué
yogué
ahí
ají
allí
aprendí
aquí
belroquí
carmesí
construí
chagüí
destruí
encogí
engullí
erguí
esquí
exhibí
frenesí

guiguí
instruí
kasrocí
kolibrí
konseguí
koquí
leí
maniquí
marroquí
partí
permanecí
perseguí
recaí
rubí
usraguí
greíl
jugnaíl
ñegneíl
wocdoíl
baúl
gleúl
raúl
jugoúl
arrojarán
callarán
capellán
escucharán
guardián
güecdumán
pakistán
posromán
zaguán

andén
daquén
guarén
jugnuclén
ñoctuguén
también
vichuquén
wognukén
güillín
jardín
jazmín
joakín
jollín
kagüín
ñujucdín
wogneguín
aberración
advección
abyección
acción
aflicción
coacción
cohesión
cohibición
crucifixión
erupción
exacción
excepción
fanfarrón
fluctuación
guión
icneumón

ignición
inyección
klistrón
kocción
konexión
mejillón
narración
neón
proyección
reglón
resurrección
sinrazón
subducción
vegetación
algún
aún
camerún
guillatún
juanrún
runrún
según
wognún
absorbió
acentuó
accesó
aglutinó
alrevesó
apoyó
asignó
bajó
desriñonó
desranchó
desrobló
desvirtuó
enrevesó
empolló
enyesó
escrudriñó
evacuó
eximió
explayó
extractó
fomentó
fondeó
konsignó
pasmó
pateó
perpetuó
quedó
raptó
relampagueó
soltó
subrayó
sucreñó
freír
oír
reír
sonreír
craúr
gleúr
kehúr
mohúr
tahúr
además
fingirás
jamás
juzgarás
kugnás
pagarás
plexiglás
trincharás
wisreclás
acdigüés
cienpiés
cortés
después
holandés
inglés
logroñés
yangüés
wognigués
anís
chisgarabís
hachís
jueguís
ñugnogüís
país
París
wongecrís
adiós
badajós
berrós
hipoglós
intradós
juñugós
semidiós
quijuekós
autobús
emaús
felús
glacdibús
jesús
jiakagús
obús
wegiagnús
belcebú
gluglú
golrocnú
guabiyú
güinrú
iguasú
konrudú
magnuruyú
ñandú
woccioctú
groíz
maíz
raíz
wocteíz
asroúz
gleúz
gocnaúz

Graves

alcancía
arqueología
conocía
coreografía
elegía
engreída
extinguía
genealogía
geología
guía
hidalguía
icnología
ingeniería
jerarquía
nacía
orgía
radiotelefonía
tabaquería
veguería
yegüería
zoología
cloúña
continúa
desritúa
desvirtúa
exceptúa
gacitúa
goúgria
grúa
maúlla
púa
rehúsa
teúrgia
áspid
ástrad
jáclud
jácced
ñácrod
céspid
güérrod
huésped
jégnud
wénrad
wécord
íngrid
guílrod
jícrud
ñílred
wírkad
clókod
gócrid
guónrad
jóñud
wósked
clúgud
glútand
gúclid
kúcrod
ñúqued
caíste
críe
freíste
greírse
güeclíe
acentúe
evacúe
exceptúe
insinúe
jaúrce
árbol
desrátil
fácil
frágil
grácil
kárel
estéril
estiércol
fértil
imbécil
trébol
aníbal
frísol
guínjol
inverosímil
níquel
cónsul
dócil
fósil
ignóbil
móvil
dignúbil
fútbol
núbil
púgil
dúctil
túnel
desiderátum
kelrámem
mágnum
memorándum
wácdam
grécdum
jégnum
llégüem
réquiem
vademécum
wéctam
delírium
guígam
ídem
ítem
jíclem
dextrórsum
góñem
kójem
wókem
wótem
clúñem
crúxem
tracdúmem
críen
guíen
kogían
reían

traían	catéter	raíces	cádiz
aúllan	césar	sonríes	cáliz
goúrden	güilréter	bahúles	gonzález
jaúñan	héctor	grúas	lápiz
rehúsan	kéfir	ñaúgas	velásquez
reúnen	néctar	púas	alférez
abstraído	wilréker	reúnes	estévez
bohío	almíbar	ántrax	pérez
estío	clíper	ápex	vélez
extravío	elíxir	gáglix	wéñez
frío	esfínter	wágnux	domínguez
leído	menjíbar	crévox	henríquez
lío	nívar	félix	manríquez
mío	dólar	gléñux	ramírez
vacío	lóker	télex	rodríguez
búho	póker	clímax	crókez
dúo	prócer	índex	glaccióvez
evalúo	azúcar	ínix	gómez
feúcho	dúdar	wíkox	gróvez
sitúo	flúor	gónrox	lópez
ámbar	húsar	kóccix	grúñoz
cadáver	súper	ónix	gúlrez
carácter	wecdúter	tórax	faúndez
cráter	fíes	dúplex	túnez
káiser	fríes	júvox	wavúclez
magnáter	oídos	kúgnax	
mártir	países	túrmix	

2. CONJUNTO DE VALIDACIÓN

A continuación se presenta el listado de 135 palabras del conjunto de validación, separado en palabras con acento implícito y explícito, y subdividido en palabras monosílabas, agudas, graves y esdrújulas. Para representar la vocal acentuada en las palabras con acento implícito, se utiliza el tilde invertido (`).

Palabras con acento implícito

Monosílabos

dièz	piè	pàr	wàtt

Agudas

albañìl	desesperàr	matìz	tiradòr
callejeàr	flechàr	molèr	tropicàl
cañaduzàl	geniàl	oxigenàr	volàr
centràr	gibàr	quitàr	
cosèr	irracionàl	recitàr	
cucùy	jirèl	terròr	

Graves

axiòma	cuncùna	flèma	gùsto
balàzo	chanchùllo	fuìste	hechicèro
bazùca	dùras	gacèta	helècho
blèdo	efectìvo	galiàno	hòja
cañàdo	empeñàda	garbànzo	hubièras
ciència	exclaustràmos	guiñàpo	huèso
circùla	fictìcio	guìsa	ignoràmte

iguàla	lùces	plutocràcia	trayèndo
imprèso	llàna	pòmo	tràzo
ìndias	manàgua	porcìno	trèta
iràrse	mazòrca	posèes	tùmbo
ìtrio	memòria	precìso	turbìna
jaquèca	mùsa	quèque	ufàno
jàrra	ñàme	regüèldo	vacùno
jèfa	ñàña	rìña	visìta
jofàina	òca	ròce	yèsca
kùrdo	òcio	Sàrtre	zàrza
làncha	òlla	siènte	zòrra
làrgo	paràdo	sìrio	zùrdo
lèña	pequeñìto	tràma	

Palabras con acento explícito

Monosílabos

sí	él

Esdrújulas

árabes	kilómetro	plástico	volcánico
fóculo	minúsculo	prístino	
homóplato	órbita	sácala	
índigo	óxido	úrico	

Agudas

así	expectación	lapón	tisú
cayó	fricción	rocín	
delfín	guiñón	salió	

Graves

árbor	mecía	rocío
crío	nenúfar	wéber

3. CONJUNTO DE PRUEBA

A continuación se presenta el listado de 135 palabras del conjunto de prueba, separado en palabras con acento implícito y explícito, y subdividido en palabras monosílabas, agudas, graves y esdrújulas. Para representar la vocal acentuada en las palabras con acento implícito, se utiliza el tilde invertido (`).

Palabras con acento implícito

Monosílabos

frèz	rièl	sèr	kìt

Agudas

abrìl	restregàr	abigarràr	abdicàr
abucheàr	acogèr	abollàr	branquiàl
abruñàl	muimùy	abrunàl	perchèl

cafìz	fechàr	roedòr	
doquièr	hacinàr	mezcàl	
exageràr	inyectòr	chillàr	

Graves

acimbòga	guìño	vièja	cigüète
cuyàno	gùzgo	pùches	güìña
camùza	herrèro	hallàda	bròte
centillèro	hullèro	aguàcha	diàntre
quebràcho	hòrma	xenofòbia	puènte
decadència	guapèras	vidòrra	vìdrio
colùmna	huèrto	chùsma	retràta
enrùga	infànte	ñècla	hayèdo
chùncho	ijàda	ñòqui	cuàrzo
trùlla	ingènio	òbvio	callèja
bisulfìto	ìngle	ojòta	capùllo
cizàña	irritàble	ocelòte	toxìna
excluyàmos	ìctus	rapàdo	uñàdo
famìlio	jèrga	ceñìdo	verrùgo
rèma	jàrcha	elegància	zanquìta
intuìble	quèchua	bròNco	yèrba
gallèta	pàila	panchìto	zatàra
guàcho	mùcho	desèes	yòga
galàyo	tàgua	platìllo	zùrra
guizàzo	uñàdo	retòque	

Palabras con acento explícito

Monosílabos

té	más

Esdrújulas

águila	jónico	cesárea	atáxico
alófono	chúcaro	chícora	
hórrido	órdiga	máximum	
íncola	ónique	júpiter	

Agudas

alfaquí	expansión	lexicón	bambú
changó	fruición	adoquín	
alhelí	gañón	bongó	

Graves

alcázar	geología	reenvío
vahído	charrúa	wélter