

INTERFAZ VOZ-TEXTO PARA LA CONSULTA DE BASE DE DATOS PARA LA TRADUCCIÓN AL LENGUAJE DE SEÑAS

Jaziel Villa Román, Ignacio Cruz Domínguez, V. Felipe Villa Román.

Facultad de Ingeniería y Tecnología

Universidad De Montemorelos

Montemorelos, Nuevo León, México

`jvilla@alumno.um.edu.mx, icruz@um.edu.mx, felipe.villa@bleext.com`

Resumen— *En este proyecto se llevó a cabo una integración de librerías para el desarrollo de una interfaz de traducción de voz a lenguaje de señas. Se utilizó el paquete Speech-Recognizer para la conversión voz a texto, el procesamiento del texto se llevó a cabo utilizando el librerías de Sklearn y NLTK y la salida producida proviene de un conjunto de imágenes almacenadas en una base de datos de MySQL. Éstas librerías se utilizaron sobre el lenguaje de programación Python y el sistema operativo Ubuntu 16.04.*

México y en su mayoría para otros idiomas de entrada, por lo tanto es de gran importancia desarrollar herramientas que apoyen en la comunicación en lenguaje de señas con las señas utilizadas en México.

Diversos trabajos se han realizado para propiciar una comunicación entre diferentes lenguajes, ya sea de idiomas hablados [5], de reconocimiento de señas [6], o de reconocimiento de voz, en ellos se destaca el uso de sistemas Speech to text (SST) y los modelos ocultos de Markov y una conexión a la base de datos.

I. INTRODUCCIÓN

De acuerdo a la Organización Mundial de la Salud (OMS), se llama defecto de audición a la incapacidad de oír tan bien como un habitante cuyo sentido del oído es normal. A los habitantes que presentan este problema pueden ser duros de oído o sordas. Si el individuo no oye nada en absoluto, lo que padece es sordera [1]. En el mundo hay 360 millones de personas con pérdida de audición discapacitante [1]. Esto equivale aproximadamente al 5% de la población mundial. De los afectados, 32 millones son niños[1]. En México en el año 2010, las personas que tenían algún tipo de discapacidad eran 5 millones 739 mil 270, lo que representaban 5.1% de la población total [2].

La audición discapacitante implica diferentes desafíos en la vida cotidiana, entre los cuales se encuentra la comunicación. Las personas con discapacidad auditiva se comunican utilizando el lenguaje de señas. El lenguaje de señas presenta la principal vía de comunicación entre personas sordas y no sordas que dominan el lenguaje de señas [3]. Uno de los problemas en México es la discapacidad auditiva que conlleva desafíos no solo para las personas que la padecen [4], sino también las personas que no sufren discapacidad auditiva, pero que no conocen el lenguaje de señas enfrentan dificultades al comunicarse con personas que presentan discapacidad auditiva, de allí la importancia de medios alternativos como la traducción. En la actualidad existen traductores a señas pero el conjunto de señas en español que utilizan es diferente al de

Uno de los primeros sistemas de comunicación oral para personas sordas en España, es un sistema que para desarrollarse, primero es necesario generar una base de datos con un corpus paralelo de frases en castellano y Lengua de Signos Española (LSE). El reconocimiento de voz que se utilizó es el reconocimiento de voz continuo basado en modelos ocultos de Markov, como se muestra en la Fig. 1 [7], otro sistema fue desarrollado en el Instituto de Stanford que es capaz de involucrar a un operador humano en una conversación natural de un tema específico [8]. Otro sistema desarrollado consiste en el reconocimiento de señas, los sistemas de reconocimiento de gestos tratan con gestos temporales, posturas de las manos y entrenamiento automático [6].

El objetivo de este proyecto es traducir de voz a lenguaje de señas. Toma como entrada la voz, la convierte a texto y procesa el texto para producir una salida [8]. La salida es una interpretación con imágenes en lenguaje de señas, permitiendo que la comunicación con una persona sorda sea una actividad sencilla y rápida [9].

El objetivo del desarrollo de esta interfaz de reconocimiento de voz es que sea capaz de reconocer frases en inglés y mostrar su respectivo significado utilizando imágenes con el lenguaje de señas de México.

La integración de diferentes librerías permite construir una interfaz de traducción a lenguaje de señas que puede lograr la

comunicación entre un persona sin discapacidad auditiva y una persona sorda.

Para el desarrollo de esta interfaz se utiliza el lenguaje de programación Python, el cual es un lenguaje de propósito general, que tiene una sintaxis fácil de leer y proporciona una retroalimentación inmediata [10] y es un lenguaje dinámico de alto nivel, de código abierto y gratuito [11], es fácil de aprender y produce un código más legible que la mayoría de los lenguajes[12].

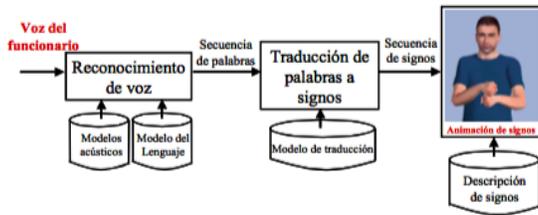


Fig. 1. Modelos principales de reconocimiento de voz [7].

II. METODOLOGÍA

Para la realización del proyecto se utilizó Python 2.7 en Ubuntu 16.4. Para reconocer la voz se utilizó SpeechRecognition 3.6.3, es la biblioteca para realizar reconocimiento de voz, con soporte para varios motores y API.

Una vez probado y logrado el reconocimiento de voz y obtenido el texto, Se instalo las librería NLTK Y Sklear para analizar el texto obtenido [13]. Se utilizó la librería de Tokenizer de NLTK para analizar y separar las palabras del texto generado por el Speech recognition, con la librería de Stemmer de NLTK se obtuvo la raíz de cada palabra del texto. Los Stemmers eliminan los afijos morfológicos de las palabras, dejando sólo la palabra raíz [13].

La clase DictVectorizer de Sklearn puede utilizarse para convertir matrices de características representadas como listas en Python. DictVectorizer sirve para realizar entrenamientos supervisados para la clasificación de modelos de procesamiento de lenguaje natural que normalmente trabajan extrayendo palabra particulares de interés [14]. La librería de NLTK ayuda a analizar el texto obtenido mediante Speech Recognition.

Para crear el modelo de entrenamiento se utilizo un Dataset, con frases relacionadas con el clima (caliente o frio) y a cada frase se asignó una etiqueta.

Se creó una base de datos para guardar las imágenes. En la consola ejecutamos el siguiente comando en terminal para instalar MySQL \$sudo apt-get install mysql-server, durante la

instalación, el servidor pedirá la selección y confirmación de la contraseña para el usuario root de MySQL.

Para conectar a al base de datos se logra con el siguiente comando en terminal, \$mysql -u root -p. Una vez conectado a MySQL se creó la base de datos y la tabla donde se almacenarán las imágenes con tres campos diferentes, un identificador, palabra y directorio raíz de la imagen tal y como se muestra en la Fig. 2.

Para conectar la base de datos desde Python se utilizó MySQLdb, es una paquetería para la conexión a un servidor de base de datos MySQL. Para instalar el modulo MySQLdb, se ejecutó el comando, \$ pip install MySQL-python.

Para conectar la base de datos en MySQL desde Python debe asegurarse de cumplir con lo siguiente:

- Haber creado una base de datos.
- Haber creado un usuario de conexión que posea las siguientes características: usuario y clave.

```
mysql> create database imgdb;
mysql> use imgdb

CREATE TABLE pimng (id INT NOT NULL PRIMARY KEY
AUTO_INCREMENT, palabra CHAR(25), fuente
CHAR(100));

INSERT INTO pimng (id, palabra, fuente) VALUES (NULL,
```

Fig. 2. Creación de base de datos y la tabla para agregar las imagenes.

Para lograr la conexión entre Python y MySQL ejecutamos los siguientes comandos como se muestra en la Fig. 3.

```
import MySQLdb
from PIL import Image

db = MySQLdb.connect(host="localhost", db="imgdb",
user="root", password="mipassword")

cur = db.cursor()

for row in cur.fetchall():
    imgs = row[0]
    db.close()
```

Fig. 3. Comandos para conectar la base de datos y Python.

Una vez realizada la conexión a la base de datos desde Python, se crea un objeto Cursor. Esto permite ejecutar todas las consultas requeridas, como se muestra en la Fig. 4. Ahora se puede utilizar todas las sentencias SQL que sean necesarias, ejemplo: `cur.execute(SELECT fuente FROM pp WHERE palabra = "+targets[best_i]+")`.

```

if(best_dist < 0.8):
    targets[best_i] de la BD
    print("recuperando la frase %s" % targets[best_i])
    words = targets[best_i].lower()
    cur = db.cursor()
    cur.execute("SELECT fuente FROM pp WHERE
palabra = '"+targets[best_i]+'")
    data = cur.fetchall()
    if data :
        for row in data:
            imgs = row[0]
            img = Image.open(imgs)
            img.show()
db.close()

```

Fig.4. Código para mostrar imágenes conectado a la base de datos con el reconocimiento de voz.

Para mostrar imagen se instaló la librería de Pillow con el comando desde la terminal, `$sudo pip install Pillow`.

El algoritmo `word2vec` de Sklearn, su función es comparar la frase de entrada con el modelo de entrenamiento que se generó y busca la similitud de la frase con una de las frases del modelo de entrenamiento, la frase que sea más cercana al modelo de entrenamiento nos regresa la etiqueta que se agregó a la frase del entrenamiento y se guarda en una variable `targets[best_i]`, con esa etiqueta se compara con el campo `palabra` de la tabla en la base de datos como se muestra en la Fig. 4, y finalmente nos da como resultado una salida de imágenes representando la frase con señas como se muestra en la Fig. 5.

III. RESULTADOS

Se escogió un tema en específico para realizar la traducción a lenguaje de señas, en este caso solo fue el tema del clima, frío o calor (verbos en pasado, presente y futuro) por ejemplo: Mañana por la noche hará calor. Al ejecutar la interfaz, el usuario proporcionó una frase en inglés, reconociendo la voz y mostraba una imagen con señas representando la frase que se dijo tal y como se muestra en la Fig.5.



Fig. 5. El resultado de una frase en lenguaje de señas.

IV. CONCLUSIÓN Y DISCUSIÓN

En este proyecto se creó una interfaz que ayuda a lograr una comunicación entre una persona sin discapacidad auditiva y una persona sorda, la utilización de la base de datos nos permite almacenar más imágenes y extender el tema.

Las ventajas del software desarrollado son las siguientes:

- Las herramientas utilizadas son de código abierto en comparación con otros proyectos que son de paga.
- Se puede agregar más imágenes a la base de datos para agregar más frases de temas y traducir a lenguaje de señas.

En las siguientes versiones se puede extender la traducción aumentando el tamaño de vocabulario en lenguaje de señas.

Referencias:

- [1] Organizaci6n Mundial de la Salud, "OMS | Sordera y p6rdida de la audici6n," WHO, 2015. [Online]. Available: <http://www.who.int/mediacentre/factsheets/fs300/es/>. [Accessed: 06-Dec-2016].
- [2] INEGI, "Poblaci6n. Discapacidad." [Online]. Available: <http://cuentame.inegi.org.mx/poblacion/discapacidad.aspx?tema=P>.
- [3] M. . Rodrguez Gonzlez, "Lenguaje de signos," *Biblioteca Virtual Miguel de Cervantes*, 2015. [Online]. Available: http://www.cervantesvirtual.com/obra-visor/lenguaje-de-signos--0/html/ffbeaf86-82b1-11df-acc7-002185ce6064_5.html. [Accessed: 10-May-2017].
- [4] C. Musselman, "How Do Children Who Can't Hear Learn to Read an Alphabetic Script? A Review of the Literature on Reading and Deafness," *J. Deaf Stud. Deaf Educ.*, vol. 5, no. 1, pp. 9–31, 2000.
- [5] M. Dua, R. K. Aggarwal, V. Kadyan, and S. Dua, "Punjabi Speech to Text System for Connected Words," pp. 349–355, 2012.
- [6] I. Conference and I. C. Applications, "2014

- International Conference on Intelligent Computing Applications MOBILE TRANSLATION SYSTEM FROM SPEECH LANGUAGE TO HAND,” pp. 411–415, 2014.
- [7] E. Rydberg, “Evaluating a Speech Communication System for Deaf People,” vol. 13, no. 4, pp. 327–345, 2011.
- [8] D. E. Walker, “Speech Understanding Through Syntactic and Semantic Analysis,” *IEEE Trans. Comput.*, vol. C-25, no. 4, pp. 432–439, 1976.
- [9] S. Contreras and F. De Rosa, “Aplicación de Deep Learning en Robótica Móvil para Exploración y Reconocimiento de Objetos basados en Imágenes,” in *Computing Conference (CCC), 2016 IEEE 11th Colombian*, 2016, p. 8.
- [10] M. Ateeq, H. Habib, A. Umer, and M. Ul Rehman, “C++ or python?: Which one to begin with: A learner’s perspective,” *Proc. - 2014 Int. Conf. Teach. Learn. Comput. Eng. LATICE 2014*, pp. 64–69, 2014.
- [11] Karlijn Willems, “BeginnersGuide - Python Wiki,” *Python*, 2017. [Online]. Available: <https://wiki.python.org/moin/BeginnersGuide>. [Accessed: 30-Apr-2017].
- [12] K. Jaan, *Numerical Methods in Engineering with Python 3*, 3, revisad ed. Cambridge University Press, 2013.
- [13] E. L. and E. K. Bird, Steven, “Natural Language Toolkit — NLTK 3.0 documentation,” *O’Reilly Media Inc.*, 2009. [Online]. Available: <http://www.nltk.org/>. [Accessed: 08-May-2017].
- [14] SCikitLearn, “Extracción de características - scikit-learn 0.18.1 documentación,” *ScikitLearn*, 2010. [Online]. Available: http://scikit-learn.org/stable/modules/feature_extraction.html. [Accessed: 15-May-2017].