UNIVERSIDAD DE BUENOS AIRES
Facultad de Ciencias Exactas y Naturales
Departamento de Computación

Asignación no supervisada de entonación para un sistema de síntesis del habla

Lautaro Dolberg

Director: Dr. Agustín Gravano

Buenos Aires, Mayo de 2011.

Resumen

La naturalidad del habla sintetizada por un sistema TTS (text-to-speech) depende principalmente de la entonación elegida para realizar la síntesis. A partir del análisis del texto puede extraerse información que se utiliza para predecir automáticamente la entonación adecuada. La tesis consiste en atacar dicho problema dentro de un marco acotado. Se trabaja con el idioma castellano de Argentina en oraciones declarativas. Dada una oración o frase escrita en lenguaje natural, se busca encontrar un contorno entonacional adecuado para que un sistema de TTS hipotético mejore la naturalidad de su síntesis. La tesis propone mediante la abstracción de aspectos sintácticos y acústicos un esquema de clustering, para luego encontrar una relación entre ambos grupos de clusters. Una vez encontrada la relación, se propone un proceso de predicción de la entonación, basado en la relación entre los clusters sintácticos y acústicos: para una frase a sintetizar se busca un representante sintáctico similar y se predice la entonación en base al representante acústico relacionado. El trabajo incluye una evaluación objetiva de los resultados contra dos sistemas baseline.

Índice

1. Introducción **1**

1.1. Trabajo previo . . . 1
1.2. Objetivos . . . 2
1.3. Estructura de la tesis . . . 2
1.4. Contribuciones . . . 3
1.5. Corpus . . . 3

2. Modelo **6**

2.1. Segmentación de oraciones en unidades sintácticas . . . 6
2.2. Criterio de similitud . . . 7
2.3. Modelo acústico . . . 8
2.3.1. Distancia acústica . . . 8
2.3.2. Distancia acústica punto por punto . . . 9
2.3.3. Distancia acústica basada en el grado de correlación . . . 10
2.3.4. Distancia acústica punto a punto mejorada . . . 12
2.4. Modelo sintáctico . . . 12
2.4.1. Modelo de Espacio Vectorial . . . 15
2.5. Uniendo ambos Modelos . . . 19
2.6. Evaluación del matching . . . 21
2.7. Elección de un representante . . . 23
2.8. Predicción de F_0 . . . 23

3. Desarrollo **25**

3.1. Modelo acústico . . . 25
3.2. Modelo vectorial . . . 26
3.2.1. Asignación de pesos en los features . . . 26
3.3. Clustering . . . 27
3.3.1. Integración con Weka . . . 27
3.3.2. Métodos de clustering aglomerativo . . . 28
3.4. En busca de la configuración óptima . . . 31
3.4.1. Elección de la metaheurística . . . 33
3.4.2. Greedy randomized adaptive search procedure . . . 34
3.4.3. Elección de los parámetros . . . 35
3.4.4. Detalles de implementación . . . 36
3.5. Sistema de predicción de F_0 . . . 37

4. Resultados **38**
4.1. Clustering . . . 38
4.1.1. Soluciones representativas de las configuraciones obtenidas . . . 38
4.1.2. Performance de la metaheurística . . . 40
4.1.3. Correlación de los modelos . . . 41
4.2. Sistema de predicción de F_0 . . . 41
4.2.1. Baselines y Gold standard . . . 42
4.2.2. Resultados representativos de la performance del sistema . . . 42
4.2.3. Evaluación numérica de los sistemas de predicción . . . 45
4.2.4. Escalabilidad . . . 47

5. Conclusiones y Trabajo futuro **48**
5.1. Conclusiones . . . 48
5.1.1. Clustering . . . 48
5.1.2. Metaheurística . . . 48
5.1.3. Predicción de F_0 . . . 49
5.2. Trabajo futuro . . . 50

A. Valores Experimentales **52**
A.1. Modelo Acústico . . . 52
A.1.1. Similitud entre oraciones VS Similitud entre unidades . . . 52
A.2. Modelo sintáctico . . . 55
A.2.1. Alineación de las anotaciones . . . 55
A.3. Clustering . . . 55

1. Introducción

La síntesis del habla es la producción artificial del habla humana. Se denomina "texto a habla" (TTS, del inglés text-to-speech) al proceso de implementar la síntesis del habla con un sistema basado en software o hardware. Los sistemas de texto a habla más difundidos son simplemente sistemas de concatenación que sintetizan habla concatenando pequeñas unidades fonológicas, dando como resultado sonidos en general inteligibles pero carentes de naturalidad.

Explicar la elección que efectúa un hablante para asignar un contorno entonacional[1] a cada oración es un tema de suma importancia en Lingüística y Fonología, así como en aplicaciones tales como sistemas de TTS. La mayoría de los sistemas actuales de TTS realizan análisis sobre atributos léxicos, sintácticos y morfológicos del texto para determinar la prosodia a utilizar. Estos sistemas normalmente logran un buen desempeño en una oración o frase aislada; sin embargo, no consiguen buenos resultados cuando se trata de un discurso prolongado.

El trabajo realizado en esta tesis consiste en atacar dicho problema dentro de un marco acotado. Se trabajará con el idioma castellano de Argentina en oraciones declarativas, ya que no existen sistemas disponibles para dicho idioma. Dada una oración o frase escrita en lenguaje natural, en castellano de Argentina, se buscará encontrar la entonación adecuada. A futuro, estos resultados podrán ser utilizados en un sistema de TTS que se encuentra en desarrollo en el Laboratorio de Investigaciones Sensoriales, Hospital de Clínicas, Universidad de Buenos Aires.

1.1. Trabajo previo

Habitualmente los sistemas de asignación o predicción de prosodia para TTS lo hacen a partir de características léxicas, morfológicas, sintácticas y contextuales del texto de entrada, siendo entrenados en forma supervisada. Pueden agruparse en dos grandes familias:

1. **Predicción de categorías prosódicas:** Estos sistemas se basan en predecir los eventos prosódicos más relevantes (por ejemplo, en ToBI: acentos tonales, acentos de finales de frase y niveles de juntura entre palabras [BH94, PBH94]).

2. Predicción de parámetros acústicos, como por ejemplo la predicción del contorno entonacional o el modelo de Fujisaki [FH84].

Los sistemas de la primera familia (por ejemplo, [Hir93, RO96, Nak98, PH00, Sun02, Spr94]) tienen como ventaja que emplean una representación compacta de los eventos prosódicos relevantes. La forma de representar dichos eventos es mediante anotaciones de acentos tonales, acentos de finales de frase, entre otros. Las anotaciones cumplen un rol clave y la problemática se da porque esta información debe ser generada por personas calificadas; además debe considerase que la asignación de etiquetas es susceptible al oyente que realiza la tarea, por lo tanto resulta una actividad costosa en términos temporales y de recursos humanos. Otro punto en contra es que

[1] Contorno entonacional: Contorno que tiene la frecuencia fundamental proveniente del habla humana para una producción dada.

no es directo el paso entre eventos discretos prosódicos a la frecuencia fundamental, es necesario dar un salto más, que no siempre viene acompañado de resultados exitosos.

Los sistemas de la segunda familia (por ejemplo, [Tor08, Fuj94, Fuj02, Mix95, Mix00]) tiene como ventaja que trabaja directamente sobre representaciones continuas de la frecuencia fundamental, evitando el salto de categorías discretas a frecuencia fundamental que hay que dar al final de la predicción con el grupo 1. Como desventaja tiene que ajustar correctamente los modelos numéricos a las grabaciones del corpus de entrenamiento tiene un margen de error no despreciable, que llevaría a un impacto negativo en los resultados finales de la síntesis. En esta tesis se explora la factibilidad de un nuevo enfoque para la predicción de F0.

1.2. Objetivos

El objetivo de esta tesis es predecir la frecuencia fundamental[2] de una frase declarativa en lengua castellana mediante un sistema no supervisado de predicción. La frecuencia fundamental es un factor clave en los sistemas de TTS para lograr un sonido que resulte natural al oyente.

Para el desarrollo del sistema se analizó un cuerpo de datos de oraciones aisladas en castellano de Argentina, descripto en la sección 1.5. Este trabajo se basa en la hipótesis de que existe una correlación positiva entre los aspectos sintácticos y los aspectos acústicos de una frase. El trabajo comienza con la definición de un modelo sintáctico y un modelo acústico, con el fin de agrupar los elementos del corpus basándose en su similitud acústica y sintáctica. El método para agrupar los elementos es mediante clustering, una técnica no supervisada de machine learning. Una vez obtenidos los clusters acústicos y sintácticos se establece una relación o matching entre los clusters, que permite relacionar cada cluster sintáctico con un cluster acústico. Para predecir la frecuencia fundamental de una nueva frase se determina el cluster sintáctico correspondiente a la frase nueva, y se le asigna la frecuencia fundamental del representante acústico con el cual se relaciona a través del matching. El esquema de la figura 1 resume el funcionamiento general del sistema de predicción de F_0. Una vez desarrollado el sistema se evaluó el rendimiento de las predicciones, comparando el sistema con dos sistemas baseline de predicción estáticos.

1.3. Estructura de la tesis

La tesis se encuentra dividida en 5 secciones y un apéndice. En la sección 2 se presenta el desarrollo de los modelos acústicos y sintácticos, el modelo de clustering jerárquico y los criterios para determinar una relación o matching entre los clusters sintácticos y acústicos, y finalmente se presenta el modelo de predicción de la frecuencia fundamental. En la sección 3 se explica el trabajo realizado durante la construcción del sistema y el diseño de los experimentos. En la sección 4 se presentan los resultados y en la sección 5, se discuten los resultados obtenidos junto

[2]Frecuencia Fundamental (F_0): Es la frecuencia más baja del espectro de frecuencias tal que las frecuencias dominantes pueden expresarse como múltiplos de la misma. En el habla, está fuertemente ligada a la percepción de la entonación.

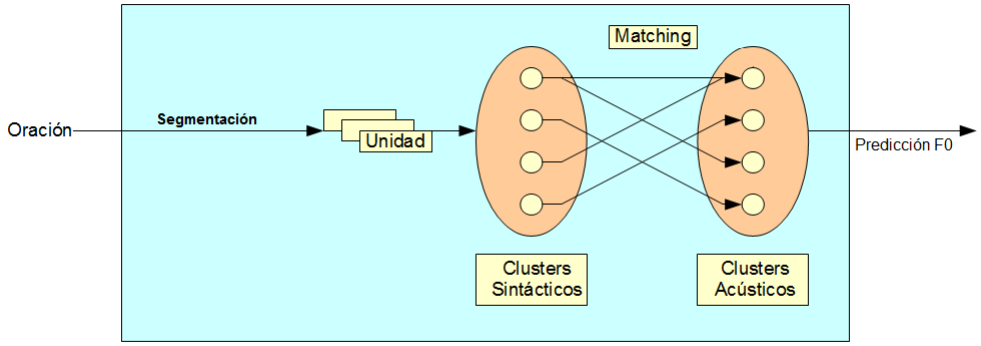


Figura 1: Esquema del sistema de predicción de F_0

al trabajo futuro. El apéndice A presenta la justificación de valores experimentales que fueron utilizados durante el desarrollo del sistema y los experimentos.

1.4. Contribuciones

La tesis aporta un nuevo enfoque a los sistemas de predicción de la frecuencia fundamental ya que hasta el momento la mayoría de estos sistemas son supervisados. A su vez el trabajo verifica la hipótesis de la existencia de una relación entre la estructura acústica y la sintáctica de una oración en lengua castellana. La tesis aporta nuevas definiciones de distancia acústica y sintáctica para enunciados y frases sintácticas que pueden ser aplicadas en otros contextos como también para extender este trabajo. Como parte de este aporte, se implementó dentro del framework de data mining Weka las distancias sintácticas y acústicas, que facilitarán en un futuro realizar nuevos análisis con esta herramienta. En general, este trabajo aporta un framework para explorar nuevas variantes del enfoque propuesto para la construcción de un sistema de TTS.

1.5. Corpus

En esta tesis se usó el cuerpo de datos SECYT de 741 enunciados, desarrollado en el Laboratorio de Investigaciones Sensoriales, Hospital de Clínicas, UBA [GRCT01]. Cada **enunciado** consiste en una oración declarativa asociada a una grabación realizada por una locutora en castellano de Argentina y a un conjunto de etiquetas que describen la acentuación, la entonación, la transcripción y la posición en el audio de cada palabra. La oración de cada enunciado puede ser segmentada en frases sintácticas que llamamos **unidades**. Para cada enunciado se cuenta con numerosas anotaciones, de las cuales usamos las siguiente en este trabajo:

1. anotación de clase de palabra (part of speech tag);
2. indicación de si la oración del enunciado es unimembre o bimembre;

3. separación en sujeto y predicado anotada a nivel de palabras.

Las clases de palabras fueron anotadas por un lingüista, usando las 19 etiquetas detalladas en la tabla 1 [TG04].

1	C	Conjunción coordinante
2	A	Preposición
3	E	Determinante
4	D	Adjetivo
5	S	Sustantivo singular
6	U	Sustantivo plural
7	T	Sustantivo propio singular
8	M	Pronombre
9	V	Adverbio
10	X	Palabra extranjera
11	N	Número cardinal
12	J	Intersección
13	I	Verbo en infinitivo
14	H	Verbo en pasado simple
15	P	Participio presente
16	B	Participio pasado
17	R	Verbo en presente
18	F	Verbo en futuro
19	O	Verbo Transitivo

Tabla 1: Part of speech tags del corpus

Frase Sintáctica o unidad Se define como unidad de análisis a la **unidad**, cada frase sintáctica en que puede segmentarse una oración.

Corpus normalizado El corpus original se adaptó para satisfacer las necesidades del modelo acústico/sintáctico. Se realizó una normalización de las unidades, es decir, transformar todos los pitch tracks[3] de cada unidad para que las mediciones se encuentren dentro del intervalo [0,1]. Esto se debió a la gran variedad de duraciones que presentaban las unidades.

Datos: Entrenamiento y Testing Parte del trabajo realizado consistió en utilizar y desarrollar técnicas basadas en algoritmos de machine learning que requieren una fase de entrenamiento. Como la performance de dichas técnicas será evaluada con datos provenientes del mismo corpus se decidió dividir el corpus en 2 secciones del mismo tamaño: la primera para realizar el entrenamiento de las técnicas desarrolladas, y la segunda para evaluar su performance.

[3]Pitch track: Es una secuencia de mediciones discretas de una onda, y representa la frecuencia fundamental. Se construye como un muestreo de una señal en un intervalo de tiempo, utilizando una ventana temporal para efectuar cada medición.

Datos: Randomización Se desconoce si el corpus fue generado en algún orden o siguiendo cierto criterio. En consecuencia, se re ordenaron sus enunciados al azar, para descartar cualquier posibilidad de condicionamiento basado en el orden en el cual la locutora grabó los enunciados y el corpus fue generado.

2. Modelo

Esta sección tiene como objetivo describir el modelo de información que se construyó durante el analisis del dominio y el desarrollo del trabajo. Comienza con el análisis del dominio del problema, luego el problema en sí y posteriormente con el análisis del corpus con el fin de conocer sus alcance y limitaciones.

La siguiente lista presenta las diferentes etapas que se llevaron a cabo para determinar el modelo del sistema propuesto para resolver la problemática tratada en este trabajo.

- Definición de la unidad mínima de información
- Desarrollo del Modelo acústico
- Desarrollo del Modelo sintáctico
- Clasificación y matching
- Búsqueda sistemática de los parámetros óptimos para la clustering y matching
- Desarrollo Modelo de predicción de F_0

Durante el desarrollo de esta sección se tratarán detalladamente cada uno de los items de la lista. El desarrollo de las métricas propuestas es parte del trabajo realizado.

2.1. Segmentación de oraciones en unidades sintácticas

Como ya se mencionó, el corpus cuenta con 741 unidades para las cuales se dispone la transcripción léxica, la grabación y un conjunto de etiquetas con información sintáctica y gramatical. La segmentación se realiza a partir de una oración extraída a partir de la transcripción léxica y el árbol sintáctico.

Dados una oración y su árbol, existen varias formas de segmentar la oración dada, esto se puede ver más en detalle en la figura 2. De aquí en adelante se definirá una unidad de análisis (la **unidad**), y sobre ella se hará referencia a la hora de detallar el análisis, desarrollo y resultados del trabajo. Una unidad se llama a una frase sintáctica perteneciente a una oración dada. Puede verse un ejemplo en la figura 3, cómo se segmenta una oración que corresponde a una transcripción léxica de un enunciado del corpus, de forma tal que la oración «El virus temporalmente se detiene» se divide en dos unidades, «El virus» y «Temporalmente se detiene», donde cada una de las unidades proviene de cada una de las frases sintácticas de la oración original respectivamente.

Existen otros niveles de segmentación posibles, como la segmentación mínima, que es aquella que solo toma la raíz como unidad, y la máxima, aquella que toma todas las hojas del árbol como unidades. Del cuerpo de datos no sólo es posible extraer oraciones simples de los enunciados, sino que existen enunciados cuya transcripción léxica corresponde a una oración compuesta. En caso que sea una oración compuesta, la segmentación se efectúa en cada suboración. Por ejemplo si se tiene la oración compuesta: «El virus temporalmente se detiene aunque no sabemos por cuanto

tiempo», se divide en dos sub oraciones «El virus temporalmente se detiene» y «Aunque no sabemos por cuanto tiempo», una vez dividida en oraciones individuales se aplica la segmentación ya descrita. Cuando la segmentación se hace efectiva es necesario acompañarla con información de contexto para eventualmente poder reconstruir la oración original, sobre este punto en particular se discutirá en la sección 2.4.1.

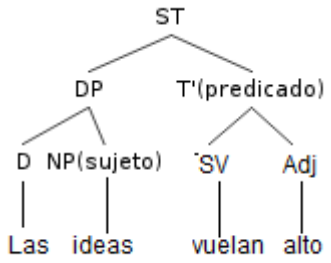


Figura 2: Ejemplo Arbol Sintáctico

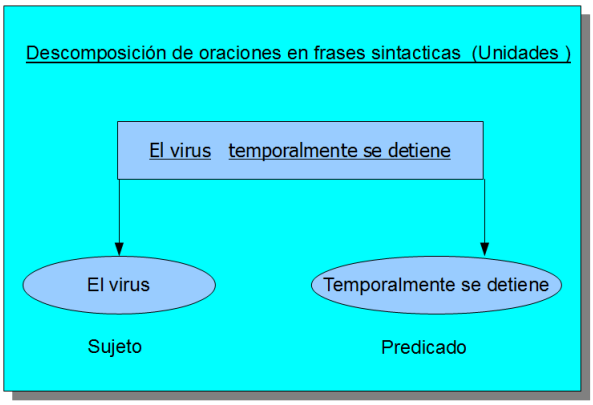


Figura 3: Ejemplo de segmentación de oraciones en unidades

2.2. Criterio de similitud

Tanto para el modelo acústico como el modelo sintáctico se definió una noción de distancia con el fin de construir clusters de unidades similares. La noción de distancia se basa en un criterio de similitud entre dos elementos dados. Con este criterio es posible agrupar los individuos de un conjunto en subconjuntos tales que la distancia entre ellos es relativamente baja. Otro punto de gran importancia en la búsqueda del criterio de similitud, es que se tiene como hipótesis que a unidades sintácticamente similares, se corresponden unidades acústicamente similares.

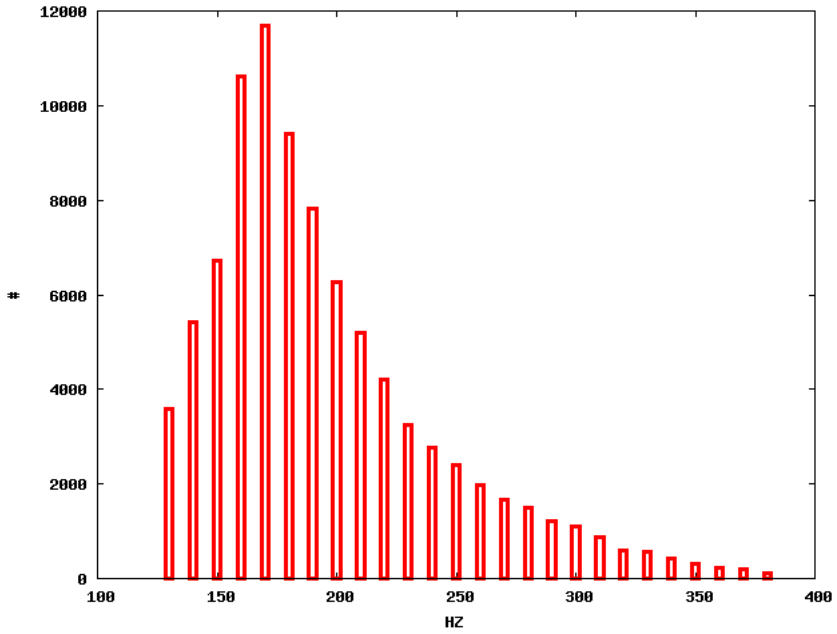


Figura 4: Histograma de las frecuencias ampliado

2.3. Modelo acústico

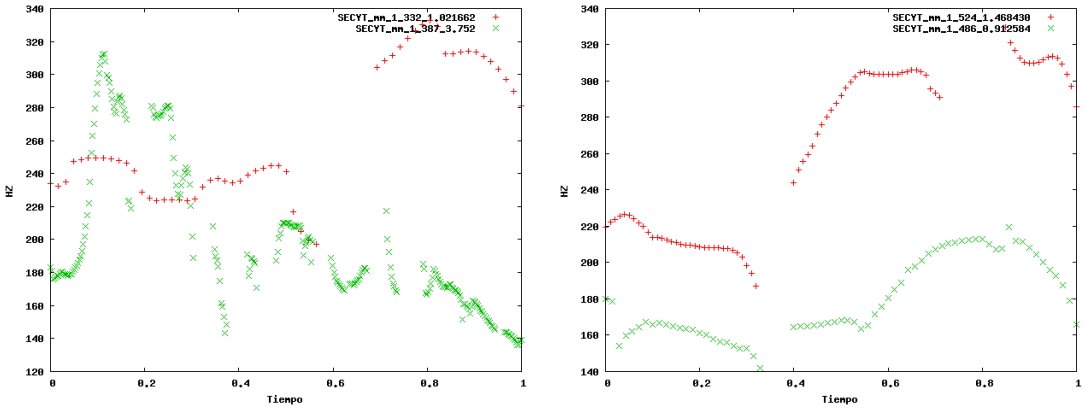
El modelo acústico se basa en el análisis del pitch track de las unidades, siendo este el único aspecto prosódico utilizado para construir el modelo para el análisis de las unidades. Para cada enunciado del corpus se encuentra disponible un archivo con una función discreta tal que el eje x representa el tiempo y el eje y, mediciones de la frecuencia fundamental del habla correspondiente medida en Hz. La frecuencia fundamental F_0, es la frecuencia más baja del espectro de frecuencias tal que las frecuencias dominantes pueden expresarse como múltiplos de la misma.

En la figura 4 puede verse un histograma de la siguiente forma: el eje x es el valor medido de la frecuencia del tono medido en Hz y el eje y es la cantidad de mediciones en todos los archivos del corpus que tienen valor x. A partir de dicho histograma se puede deducir que existe un rango acotado de entre 130 y 380 Hz para la F_0 de la hablante que ha grabado las unidades del corpus.

2.3.1. Distancia acústica

Intuitivamente dados dos sonidos podemos con sólo escucharlos determinar si sus entonaciones son similares. Por ejemplo, si se escuchan dos sonidos, se puede identificar si ambos fueron graves o agudos, si hubo cambios en el tono durante el transcurso de los mismos. La onda que resulta al graficar el pitch track se considera una función f:$\Re \times \Re$. Dadas funciones de pitch track, f_1 y f_2, es necesario determinar cuándo son similares, en forma similar al oído humano. Si se observa la figura 5(a) puede verse el pitch track que corresponde al audio SECYT_mm_1_332 comienza

con una frecuencia de aproximadamente 240Hz, luego desciende dos veces en el intervalo total de tiempo, para finalizar más alto de lo que comenzó. En cambio en la figura 5(b) puede verse el pitch track del archivo SECYT_mm_1_387 tiene sucesivos altos y bajos en el total del intervalo temporal y luego finaliza con un tono muy grave. Así como es posible encontrar dos elementos que presenten severas diferencias, es posible encontrar elementos del corpus con un gran parecido, como puede verse por ejemplo en la figura 5(b), donde ambos contornos entonacionales trazan una curva similar, aunque a diferente frecuencia.



(a) (b)

Figura 5: 5(a) Contornos entonacionales con severas diferencias correspondientes a dos oraciones en la base de datos, 5(b) Contornos entonacionales con un gran grado de similitud correspondientes a dos oraciones en la base de datos

Se define **la distancia acústica** como una función

$$D(f_1, f_2) : (\Re \to \Re) \times (\Re \to \Re) \to \Re$$

que dados dos pitch tracks normalizados para sendas unidades, f_1 y f_2, proporciona una noción de similitud entre f_1 y f_2. Intuitivamente se explica como el parecido entre los contornos entonacionales de la funciones que f_1 y f_2 describen. A continuación se describen tres variantes estudiadas para modelar la distancia acústica entre dos unidades cualesquiera del corpus.

2.3.2. Distancia acústica punto por punto

La primera definición de distancia contemplada consiste en la suma de norma 2 entre todo par de puntos de los pitch tracks

$$D(f_1, f_2) = \sum_{x \in [0..1]} |f_1(x) - f_2(x)|$$

donde x pertenece al intervalo de números reales entre 0 y 1 para los cuales existe un registro de F_0 en el pitch track normalizado de la unidad.

Cabe destacar que ambas funciones son una discretización del pitch track, esto implica que dado un $x_0 \in [0..t_{f_1}]$, es decir un instante de tiempo en la duración de la oración para la cual existe un registro de tono, es posible que no exista un mismo instante de tiempo en el dominio de f_2 que coincida con x_0. En dicho caso se procede a realizar la siguiente operación: Se busca determinar la proporción de mediciones que se dispone, de forma tal que pueda compararse punto a punto. La proporción está dada por el cociente entre la cantidad de mediciones de los pitch tracks. Por ejemplo si uno de los pitch track esta compuesto por el doble de mediciones que el otro, cada punto del primero se usará dos veces por cada punto del que menos mediciones tiene. La función de distancia descrita tiene las siguientes ventajas:

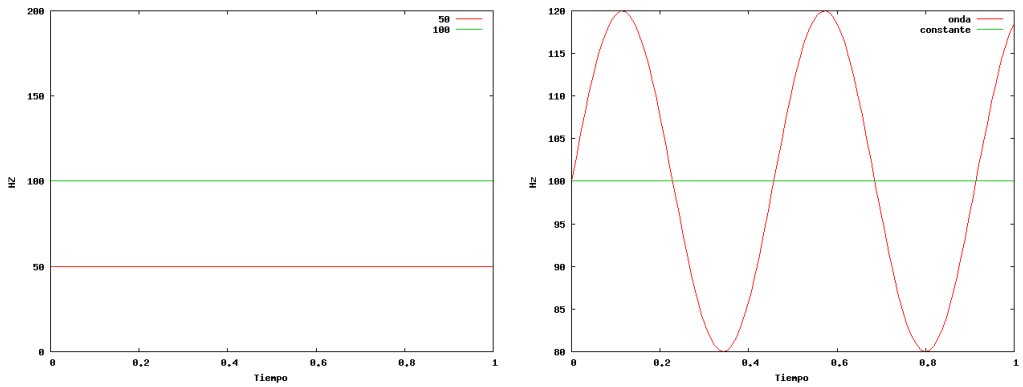
- su implementación es sencilla en casi cualquier lenguaje de programación y es parte de muchas librerías standard;
- es de orden lineal respecto al tamaño de la entrada, la cantidad de mediciones que hay en el intervalo normalizado [0..1];
- es numéricamente estable ya que para el problema actual no se cae en casos numéricamente problemáticos.

Sin embargo, no es lo suficientemente precisa para poder determinar que realmente exista una similitud prosódica entre dos unidades. Esta situación se ve ejemplificada en la figura 6(a) en la cual se observan 2 funciones constantes, cuyo contorno es similar, salvo por un desplazamiento constante que las separa. Sin embargo, la función de distancia computará un valor bastante grande, ya que para cada punto la distancia entre ambas funciones constantes es de 50 Hz. Esto representa un caso donde se espera que la función de distancia sea mínima y no sucede. A su vez puede verse el caso ejemplificado en la figura 6(b), donde al computar la función de distancia se obtendrá un valor muy chico, dado que en cada punto, la diferencia entre ambas funciones es de 20 Hz como máximo. Intuitivamente se esperaría que la distancia acústica del primer caso fuer menor que la del segundo, y vemos que con esta definición de distancia no ocurre así.

2.3.3. Distancia acústica basada en el grado de correlación

La segunda opción analizada consiste en la similitud entre dos unidades como el grado de independencia que existe entre f_1 y f_2, las funciones que representan los pitch tracks a comparar. Una métrica posible para ver cuán correlacionadas están ambas mediciones es la covarianza, definida como:

$$S_{xy} = E([X - E(Y)]E(Y - E(X))]$$



(a) (b)

Figura 6: En este gráfico puede verse dos ejemplos en los cuales la función de distancia punto a punto presenta puntos de falla al indicar la similitud del contorno entonacional

donde E() es la esperanza. Sin embargo para distribuciones discretas puede usarse la formula:

$$S_{xy} = \frac{1}{n}\sum_{i=1}^{n}(x_i - \overline{x})(y_i - \overline{y})$$

La covarianza indica el grado de dispersión teniendo en cuenta que:

- Si $S_{xy} > 0$, hay dependencia directa (positiva), es decir, a grandes valores de x corresponden grandes valores de y.
- Si $S_{xy} = 0$, no existe una relación lineal entre las dos variables estudiadas.
- Si $S_{xy} < 0$, hay dependencia inversa o negativa, es decir, a grandes valores de x corresponden pequeños valores de y.

Con lo cual en el problema estudiado, si se desea que una función de distancia se comporte como tal, cuando $S_{xy} = 0$ entonces ambas mediciones corresponden a unidades con un contorno entonacional muy diferente. Lo mismo sucederá para valores de $S_{xy} < 0$, con lo cual si se busca que los enunciados representados por f_1 y f_2 sean similares, esto estará dado en parte por $S_{xy} > 0$. Si bien la medida de correlación entre unidades tiene propiedades favorables, como la simetría, no es suficiente para establecer una noción de distancia sólida. No es posible utilizarla como función de distancia debido a que no siempre es positiva.

Debido a que esta técnica no proporcionó grandes mejoras ni información relevante para resolver la problemática planteada, fue descartada. Sin embargo, algunos criterios extraídos de esta propuesta son utilizados como base para desarrollar la función de distancia mejorada que se detalla a continuación.

2.3.4. Distancia acústica punto a punto mejorada

Con el objeto de mejorar los dos enfoques planteados previamente, es posible combinar ambas ideas y elaborar una nueva métrica. La idea de base de la métrica mejorada es estudiar la variación de la distancia que separa ambas funciones. Por ejemplo, en el caso de las funciones constantes en la figura 6(a), la distancia que las separa se mantiene constante durante todo el intervalo de tiempo para el que existen mediciones. Sin embargo en el caso las funciones en la figura 6(b) no es tan sencillo detectar este fenómeno, ya que si se analiza todo el pitch track o si se analiza un intervalo, los resultados varían. Al analizar cada uno de los intervalos, puede observarse que la diferencia entre los contornos es notable. Es por ello que se decidió analizar los contornos entonacionales a una escala menor. Dada una función que representa el pitch track, se desea encontrar los intervalos donde es suave y sin saltos, para buscar dentro cada intervalo la variación de la distancia.

El método para calcular la distancia entre ambos contornos entonacionales, comienza separando sendos contornos en intervalos. Una vez que se han segmentado los contornos, se computa la variación de la distancia dentro de cada uno de los segmentos. Para comparar dos puntos del pitch track se usa la misma noción de distancia que en la similitud punto a punto, pero se incorpora una nueva componente para poder analizar la variación de la distancia: la distancia promedio del intervalo. Con este criterio lo que se busca es lograr que ambos contornos entonacionales tengan una curva similar, tolerando un corrimiento sobre el eje y. Esto permite que tengan parámetros prosódicos similares, pero que los contornos comiencen más grave o más agudo. La distancia promedio del intervalo se define como

$$\overline{d_{f_1 f_2}}(x) = \frac{\sum_{y \,\in\, int(x)} d(f_1(y), f_2(y))}{|int(x)|}$$

donde $int(x)$ es un conjunto de puntos alrededor de x para los cuales la diferencia entre f_1 y f_2 es menor a una constante establecida empíricamente. Entonces la distancia acústica punto a punto mejorada queda definida de esta manera:

$$D(f_1, f_2) = \sum_{x}^{[0..1]} |\overline{d_{f_1 f_2}}(x) - d(f_1(x), f_2(x))|$$

donde $d(x, y) = \sqrt{(x-y)^2}$.

2.4. Modelo sintáctico

Como se mencionó anteriormente el corpus está conformado por enunciados, una serie de anotaciones y transcripciones léxicas que acompañan a cada una de las grabaciones. Utilizando la información disponible en el corpus es posible abstraer cada unidad como un conjunto de etiquetas formado por elementos de:

- anotación de clase de palabra
- indicación de si la oración es unimembre o bimembre
- separación en sujeto y predicado anotada a nivel de palabras

Anotación de clase de palabra (Part-of-Speech Tag) Uno de los elementos importantes en todo sistema de procesamiento de lenguaje natural son las etiquetas de clase de palabra (en inglés Part-of-Speech Tag).

En la lengua castellana una palabra puede tener asociada más de una clase de palabra, por ejemplo, la palabra «nada», puede ser un verbo o un sustantivo. En el trabajo actual el etiquetado viene dado como parte del corpus y se asume correcto. De todas formas es importante recalcar que dicho proceso no es trivial y usualmente es realizado automáticamente [Bri92] [R$^+$96].

Para la lengua castellana existen pocas herramientas específicas de anotación de clase de palabra por palabra o por frase. Dentro del corpus es posible encontrar los POS Tags con la siguiente estructura, dada una transcripción léxica de un enunciado, por ejemplo a continuación se muestran la información disponible para "Alvarez se hab ía animado a contarle un chiste":

```
Tiempo Color Label
0.728914  121 Alvarez
1.322510  121 /p
1.390507  121 se
1.623004  121 había
2.046500  121 animado
2.736911  121 a
2.822149  121 contarle
3.453146  121 un
3.678731  121 chiste
4.211128  121 /p
```

La anotación de clase de palabra por palabra es presentada de la siguiente forma (POS Tags):

```
Tiempo Color Label
0.727866  121 S
1.393537  121 L
1.623004  121 O
2.047269  121 P
2.736911  121 A
2.835884  121 I
3.453146  121 E
3.679786  121 S
```

Donde por ejemplo la etiqueta S hace referencia a la clase de palabra de los sustantivos, P a la clase de palabra de los verbos y E a la clase de palabras de los determinantes. A continuación la separación en sujeto y predicado anotada a nivel de palabras

```
Tiempo Color Label
0.674829  121 Su
1.388936  121 Pr
1.623247  121 Pr
2.055270  121 Pr
2.734414  121 Pr
2.843009  121 Pr
3.456339  121 Pr
3.679786  121 Pr
```

Usando las anotaciones disponibles en el corpus, es posible abstraer cada unidad en una estructura de datos que contenga sus atributos proporcionados por las etiquetas. Por ejemplo la abstracción a una lista de etiquetas para una oración del corpus tiene la siguiente forma: «Alvarez se había animado a contarle un chiste» $\rightarrow [S, L, O, P, A, I, E, S] \cup [Su, Pr, Pr, Pr, Pr, Pr, Pr] \cup [Bi]$ donde Su y Pr representan que la palabra en la i-ésima posición pertenece al sujeto y predicado respectivamente. La etiqueta Bi se usa para indicar en caso de que la oración sea unimembre o bimembre.

Se definen los siguientes conjuntos:

- P es el conjunto de todos los valores posibles que puede tomar una etiqueta de anotación de clase de palabra (POS Tag, ver tabla 1);
- FS es el conjunto $\{Su, Pr\}$;
- B es el conjunto $\{Un, Bi\}$.

2.4.1. Modelo de Espacio Vectorial

Se conoce como modelo de espacio vectorial a un modelo algebraico utilizado para filtrado, recuperación, indexado y cálculo de relevancia de información [SWY75] . Representa documentos escritos en lenguaje natural de una manera formal mediante el uso de vectores en un espacio lineal multidimensional. Fue usado por primera vez por el sistema SMART [Sal71] de recuperación de información (IR).

Los documentos son representados como $d_j = (w_{1,j}, w_{2,j}, ..., w_{t,j})$. Cada dimensión corresponde a un término. Si un término ocurre en el documento, su valor en el vector es distinto de 0. Existen varias formas de computar los pesos que conforman el vector; una opción común es usar esquemas de peso como tf-idf (term-frecuency indica cuantas veces un término aparece en el documento; inverse document frecuency es la inversa de la cantidad de documentos que contienen el término). Sin embargo esta definición de término depende del modelo y su aplicación. Típicamente los términos son palabras solas, pero pueden ser frases o keywords. Si las palabras son elegidas para ser términos entonces la dimensionalidad del vector es tan grande como el vocabulario del corpus.

En el modelo de espacio vectorial la relevancia de un documento frente a una búsqueda puede calcularse usando la diferencia de ángulos (basada en el coseno de esos ángulos) de cada documento respecto del vector de búsqueda. Así un valor de coseno de cero significa que la búsqueda y el documento son ortogonales el uno al otro, y eso significa que no hay coincidencia; un valor de coseno igual a uno significa que los documentos son equivalentes.

En el area de IR se utiliza este modelo para encontrar aquellos documentos que son relevantes a los criterios de una búsqueda dada. Esto se logra por medio de la construcción de un vector de búsqueda a partir de las palabras claves y luego se ordenan los documentos del corpus en función a la similitud al vector de búsqueda. En el caso particular de este trabajo es importante destacar que no se hizo un uso exhaustivo en forma directa de la funcionalidad de búsqueda que provee el modelo de espacio vectorial para documentos. En vez de proporcionar al usuario documentos relevantes a partir un vector de búsqueda compuesto por keywords, se utilizó el modelo para explotar su capacidad de ordenar los documentos según su similitud, usando las unidades de nuestro corpus como documentos. Gracias a poder conocer la similitud entre las unidades, fue posible avanzar en una etapa posterior de clustering.

Aplicación del Modelo de Espacio Vectorial La aplicación del modelo vectorial parte desde el punto en el cual las unidades son consideradas documentos. Para ello es necesario un esquema de asignación de pesos. El esquema se basa en tener en cuenta los atributos sintácticos de las unidades para modelar los vectores. En esta sección se explica la transformación detallada de una unidad a un vector.

En un primer enfoque se intentó trabajar con las transcripciones léxicas de las unidades como si fuesen documentos, es decir, aplicando el modelo de espacio vectorial directamente sobre la transcripción léxica de una unidad. Sin embargo, los resultados no fueron satisfactorios debido a la

falta de información sintáctica en tales documentos. Por ejemplo, si se tiene la oración «Alvarez se había animado a contarle un chiste» y se la compara con «Rodríguez se había animado a contarle un chiste», siendo su única diferencia el sustantivo propio que conforma el sujeto de la oración, no hay posibilidad de asignar los mismos parámetros a ambas oraciones. Es importante destacar que se trabajó buscando respaldar la hipótesis de que oraciones de características sintácticas similares se corresponden con características acústicas similares. Por lo tanto, no es posible utilizando solo las transcripciones léxicas como documentos, introducir al modelo las características sintácticas con las cuales se deseaba someter al criterio de similitud.

En una etapa posterior, se propuso aplicar el modelo de VSM (Vector Space Model) a la abstracción planteada sobre las unidades, basándose en POS Tags, tipo de frase sintáctica y pertenencia a una oración unimembre o bimembre (la abstracción en detalle se presentó en la sección 2.4

Nuevamente volviendo sobre el ejemplo anterior, a la oración «Alvarez se había animado a contarle un chiste», le corresponde la abstracción

$$t \ : \ [S, L, O, P, A, I, E, S] \cup [Su, Pr, Pr, Pr, Pr, Pr, Pr] \cup [Bi]$$

. Si tenemos en cuenta también la oración modificada «Rodríguez se había animado a contarle un chiste» se obtendrá una abstracción t' idéntica a t. A partir de este resultado se extendió el modelo de espacio vectorial. En vez de utilizar el conjunto de palabras perteneciente a la transcripción léxica de las unidades, se utilizó el conjunto de descriptores sintácticos. Los descriptores sintácticos utilizados eran parte del corpus, pero el modelo es independiente de la disponibilidad de tales datos, ya que existen herramientas para extraer los descriptores necesarios en forma automática. Volviendo al ejemplo de la oración «Alvarez se había animado a contarle un chiste», de aquí se obtienen 2 unidades, una para «Alvarez» y otra para el resto de la oración, «se había animado a contarle un chiste». El vector que representa a la primera unidad, será un vector que solo tiene un ítem distinto de 0, en particular el ítem que representa a los sustantivos. La segunda unidad tiene un vector más heterogéneo ya que existen varios POS Tags presentes en la unidad.

Otra ventaja de utilizar la abstracción como documento en el modelo es que la dimensión de los documentos queda acotada a la máxima cantidad de POS Tags que sean proporcionados por el corpus. Esto resulta ventajoso ya que los documentos largos quedan poco representados, ya que contienen pocos valores en común (un producto escalar menor y una gran dimensionalidad).

Esquema de atributos en el modelo planteado A continuación definimos formalmente la transformación efectuada sobre una oración para obtener su abstracción sintáctica. Dicha transformación es una función:

$$T : Unidad \rightarrow < \Re^{\#P+\#Pred} \times \{0, 1\} >$$

Donde P es el conjunto de POS Tags, $Pred$ indica a qué tipo de frase sintáctica pertenece la unidad , y la última dimensión corresponde a si es unimembre o bimembre. Cuando se aplica la transformación sobre una unidad, se hace uso de la información de contexto que ésta provee, la cual permite saber a qué frase sintáctica de la oración original pertenecía la unidad, y si la oración era bimembre o unimembre. De todas formas es posible aplicar la transformación a una oración completa sin pérdida de generalidad.

Es necesario definir la siguiente función para poder completar la fórmula de la transformación. $Tag_i : Unidad \to \aleph$ dada una unidad x devuelve la cantidad total de ocurrencias del tag i en x. Por ejemplo:

$$Tag_{Sustantivo}(\text{“}Mariana\ hizo\ la\ mermelada\text{”}) = 2$$

$$Tag_{Articulo}\text{“}Se\ hab\acute{i}a\ animado\ a\ contarle\ un\ chiste\text{”}) = 1$$

$FS_i : Unidad \to \aleph$ dada una unidad x devuelve la cantidad total de palabras que pertenezcan al Sujeto o al Predicado en x. Por ejemplo:

$$FS_{Sujeto}(\text{“}El\ virus\ avanza\ lentamente\text{”}) = 2$$

Finalmente $BU : Unidad \to 0, 1$ que dado un string con la transcripción léxica devuelve si la oración es unimembre o no. Por ejemplo:

$$BU(\text{“}Se\ hab\acute{i}a\ animado\ a\ contarle\ un\ chiste\text{”}) = 1$$

siendo 1 y 0 los posibles resultados respectivamente. Ahora es posible definir formalmente la transformación. Sea x una unidad, entonces

$T(x) = y \Rightarrow \forall i\ con\ 0 \leq i < \#P \Rightarrow y_i = \frac{Tag_i(x)}{|x|}$. Es decir, si se toma como ejemplo x = «Alvarez se había animado a contarle un chiste» se obtendrá un vector de la forma

$$T(\text{”}Alvarez\ se\ hab\acute{i}a\ animado\ a\ contarle\ un\ chiste\text{”}) = y$$

$$y = < \frac{Tag_{Sustantivo}(x)}{8}, \frac{Tag_{Articulo}(x)}{8}, \dots >$$

De esta forma y_i toma valores dentro del intervalo [0..1].

De una forma análoga es posible definir los siguientes valores de y $\forall i\ con\ 0 \leq i < \#Pred$ donde $y_{i+\#P} = \frac{Pred_i(x)}{|x|}$, resultando:

$$y = < \frac{Tag_{Sustantivo}(x)}{8}, \frac{Tag_{Articulo}(x)}{8}, \dots, \frac{Pred_{Sujeto}(x)}{8}, \frac{Pred_{Predicado}(x)}{8}, 1 >$$

$$y = < \frac{1}{8}, \frac{1}{8}, \dots, \frac{1}{8}, \frac{7}{8}, 1 >$$

Distancia vectorial Comparando la diferencia entre ángulos de dos vectores de documentos dados, es posible encontrar una función que indique la cercanía entre estos. En la práctica es más sencillo calcular el coseno de dicho ángulo que el ángulo en sí:

$$\cos\theta = \frac{\mathbf{d_2} \cdot \mathbf{d_1}}{\|\mathbf{d_2}\| \, \|\mathbf{d_1}\|}$$

, donde $\mathbf{d_2} \cdot \mathbf{d_1}$ es la intersección entre los documentos, calculada en forma de producto interno, y $\|\mathbf{d_i}\|$ es la norma del vector d_i. La norma se calcula con la siguiente fórmula:

$$\|\mathbf{v}\| = \sqrt{\sum_{i=1}^{n} v_i^2}$$

El valor 0 como resultado de dicha función indica que ambos vectores son ortogonales y no hay coincidencia entre ellos. El máximo valor posible es 1, en caso de que exista una coincidencia total.

Distancia de edición Se llama distancia de Levenshtein[Lev66], distancia de edición, o distancia entre palabras, al número mínimo de operaciones requeridas para transformar una cadena de caracteres en otra. Se entiende por operación: una inserción, una eliminación o una sustitución de un caracter. Esta distancia recibe ese nombre en honor al científico ruso Vladimir Levenshtein, quien la creara en 1965. Es útil en programas que determinan cuán similares son dos cadenas de caracteres, como es el caso de los correctores de ortografía. Por ejemplo, la distancia de Levenshtein entre «casa» y «calle» es de 3 porque se necesitan al menos tres ediciones elementales para cambiar uno en el otro:

1. casa → cala (sustitución de «s» por «l»)
2. cala → calla (inserción de «l» entre «l» y «a»)
3. calla → calle (sustitución de «a» por «e»)

Se le considera una generalización de la distancia de Hamming, que se usa para cadenas de la misma longitud y que solo considera como operación la sustitución.

Aplicando este concepto al modelo actual, se incorpora una nueva componente al espacio vectorial definido: la concatenación de todos los POS Tags. Redefiniendo la transformación efectuada sobre la oración es una función:

$$T : Unidad \rightarrow < \Re^{\#P+\#Pred} \times \{0,1\} \times SecuenciadePOSTags >$$

Por ejemplo veamos la transformación final sobre un elemento del corpus. Se tiene el enunciado «La ciudad vieja luce una metamorfosis», de aquí se separan 2 unidades, «La ciudad vieja» y «luce una metamorfosis». A fines de ejemplificar se toma la primera, «La ciudad vieja».

$$T(\text{“}La\ ciudad\ vieja\text{”}) = y$$

$$y =< 0, \frac{1}{3}, \frac{1}{3}, 0, 0, 0, \frac{1}{3}, 0, 0, 0, 0, 0, 0, 0, 0, \text{“}ESD\text{”} >$$

Es necesario redefinir la métrica, ya que la noción de similaridad coseno ya no se puede aplicar directamente sobre los vectores después de la nueva transformación. En su lugar la función de distancia D, dados dos vectores u y v resulta:

$$D(u, v) = \alpha D_{cos}(u, v) + \beta D_{edición}(u, v)$$

En la fórmula presentada se hace un abuso de notación ya que tanto u, como v son vectores con la componente de texto en su última posición, pero en el primer caso (distancia coseno) es ignorada por la función que calcula la similaridad coseno, y en el segundo caso, las componentes que son partes del modelo vectorial original son ignoradas análogamente por la función de distancia de edición. $\alpha\beta \in \Re$, son pesos que pueden variar, y su valuación óptima será determinada en los capítulos siguientes. Sin embargo, pueden asumirse ambos como 0.5 por defecto.

En resumen, se han establecido dos modelos que permiten abstraer propiedades sintácticas y acústicas de una unidad. Para cada uno de ellos se definió un criterio de similitud que permite ser aplicado como noción de distancia entre dos unidades cualesquiera.

2.5. Uniendo ambos Modelos

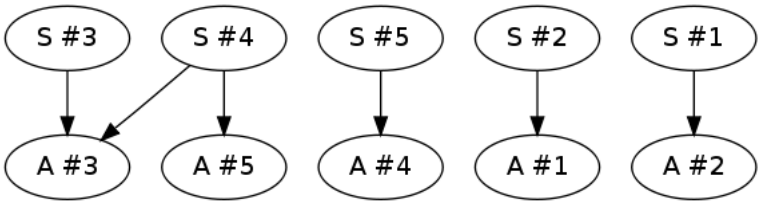


Figura 7: Ejemplo de un posible matching entre los clusters sintácticos y acústicos

Otro de los supuestos del trabajo es que es si se agrupan las unidades de acuerdo a su similitud acústica por un lado y a su similitud sintáctica por el otro, deberá existir un vínculo, o correspondencia entre ambos agrupamientos que es posible modelar mediante una relación. Entonces a continuación se buscó la forma de relacionar grupos de unidades formados basándose en su similitud sintáctica con grupos de unidades acústicamente similares. En esta sección se detalla el modelo de clustering, así como el criterio empleado para buscar y evaluar la relación entre los elementos clasificados automáticamente.

El primero de los pasos a seguir consistió en agrupar a las unidades del corpus en conjuntos tal que cada conjunto represente un cluster de individuos próximos entre sí, utilizando como noción de distancia el criterio de similitud ya desarrollado. Luego buscamos una relación entre estos

conjuntos, de forma tal que un conjunto sintáctico esté relacionado con uno o varios conjuntos acústicos. La figura 7, ilustra esto. En la capa superior puede verse la clustering sintáctica que se conecta por medio de una relación con la inferior, la acústica. En particular, en la figura 7 no hay ningún criterio en la asociación, ya que sirve a modo de ilustración.

Como solución al problema de clustering, se utilizó la técnica de clustering jerárquico [HTF01]. Esta técnica consiste en un mecanismo de búsqueda iterativa, utilizando resultados de un paso anterior, para producir el paso siguiente. Esta familia de algoritmos suele ser aglomerativa (desde abajo hacia arriba) o divisoria (desde arriba hacia abajo). La versión aglomerativa empieza con cada elemento como un cluster separado y los fusiona sucesivamente en clusters mayores. La técnica divisoria comienza con todos los elementos en un mismo cluster y divide a este iterativamente en instancias menores.

Una relación entre un agrupamiento de elementos similares acústicos y un agrupamiento de elementos sintácticos, se puede interpretar como encontrar una correspondencia o matching entre nodos de un grafo bipartito. Sin embargo, es necesario definir un criterio para establecer dicha relación. Dentro del universo de posibilidades para realizar la correspondencia se pueden plantear varias soluciones, en particular en este trabajo se efectuaron pruebas con las siguientes alternativas:

- **Elección al azar**:

  Esta solución se basa en elegir para cada conjunto sintáctico un conjunto acústico al azar, esta elección no aporta demasiada información al modelo, ya que cualquier relación es equiprobable y no tiene ningún respaldo teórico.

- **Elección basada en el algoritmo de matrimonios estables**: El problema de encontrar un Matching Estable o Matrimonio Estable entre dos grupos de individuos fue introducido por Gale y Shapley en el año 1962 [GS62]. El problema está definido de la siguiente forma: Sea un conjunto de hombres y otro de mujeres, donde cada individuo de un conjunto tiene una lista ordenada de preferencias sobre todos los individuos del otro conjunto. Si asignamos cada hombre con cada mujer tal que no exista ningún par hombre-mujer que no están juntos, pero que ambos hubieran preferido estar el uno con el otro antes que con su pareja actual, entonces ese matching es estable. Gale y Shapley mostraron que siempre existe un matching estable para cualquier conjunto de listas de preferencias, siempre que la lista de preferencias de cada agente incluya a todos los agentes del conjunto opuesto. También está demostrado que existe un matching que es óptimo para uno de los dos conjuntos en el sentido que cada agente está en su mejor asignación posible, y que si hubiera otro matching estable, estaría igual o menos conforme según sus preferencias, utilizando como preferencia el tamaño de la intersección.

- **Elección basada en maximizar una función de correspondencia:** En este caso, un matching está definido cuando se maximiza la función de forma greedy, es decir cada par

de elementos son aquellos que maximizan la siguiente función.

$$F(x, y) : Conjunto(\aleph) \times Conjunto(\aleph) \rightarrow \Re$$

$$F(x, y) = \frac{\#(x \cap y)}{\#(x \cup y)}$$

La métrica fue construida con la motivación de hallar aquellos matchings en los cuales la cantidad de unidades en la intersección de dos clusters es alta. Como ya se mencionó previamente, la hipótesis de la tesis es que existe una correlación positiva entre unidades acústicas y unidades sintácticas. Por lo tanto, los matchings en los cuales se da que la intersección entre los clusters relacionados tiene cardinalidad alta son más adecuados para representar dicha hipótesis que los matchings en los que tal propiedad no se cumple.

2.6. Evaluación del matching

Dado que más de una relación puede resultar factible al tener en cuenta los criterios previamente descritos, resulta necesario poder discernir cuál de ellas es más conveniente. Nótese que una posible configuración de conjuntos y matching podría ser, un solo conjunto en cada componente del modelo (uno con todas las unidades acústicas y otro con todas las unidades sintácticas) y una correlación trivial entre ambos conjuntos. También es posible extremar la situación pero en el otro sentido: tantos conjuntos como elementos del corpus y una relación total entre cada uno de los elementos. Sin embargo, ambos extremos no aportan información al modelo. Entonces, el criterio de **entropía de la información** resulta útil para no caer en los casos triviales en los cuales los resultados no aportan más información de la que ya se posee.

La entropía de la información fue definida por Shannon [Sha51] de la siguiente forma. Dado un mensaje se calcula para cada uno de sus símbolos:

$$I(x_i) = \log_2 \frac{1}{p(x_i)} = -\log_2 p(x_i)$$

donde $p(x_i)$ representa la frecuencia relativa del símbolo x_i. Luego para calcular la entropía del mensaje se aplica la siguiente fórmula:

$$\begin{aligned} H(X) = \mathrm{E}(I(X)) &= \sum_{i=1}^{n} p(x_i) \log_a \left(\frac{1}{p(x_i)} \right) \\ &= -\sum_{i=1}^{n} p(x_i) \log_a p(x_i) \end{aligned}$$

En nuestro problema de determinar cuánta información contiene un matching la entropía se calcula de la siguiente forma: dado un matching, como un conjunto de pares ordenados (i, j) donde i y j representan clusters sintácticos y acústicos respectivamente, el par (i, j) representa

que en la relación, el cluster sintáctico i está relacionado con el cluster acústico j. Se define: $p(x_i)$ como el porcentaje total de pares que contienen a i en la primera coordenada. En particular nos interesan aquellos matchings que poseen alta entropía ya que aquellos con valor 0, aportan muy poca información.

Un ejemplo ilustrativo consiste en tomar el matching en la figura 7 y compararlo con el siguiente matching

$$M_1 \ = \ \{(0,0),(1,0)\}$$

El matching de la figura 7 expresado en forma de relación tiene la siguiente forma:

$$M_0 \ = \ \{(1,2),(2,1),(3,3),(4,5),(4,3),(5,4)\}$$

los valores de $p(x_i)$ se encuentran calculados a continuación en la siguiente tabla:

Par	Frecuencia
$(1,x)$	$\frac{1}{5}$
$(2,x)$	$\frac{1}{5}$
$(3,x)$	$\frac{1}{5}$
$(4,x)$	$\frac{2}{5}$
$(5,x)$	$\frac{1}{5}$

Resultando la entropía del matching como el resultado del siguiente cálculo:

$$I(M_0) = \sum -log_2 p(x_i) = -log_2(\frac{1}{5}) \times \frac{4}{5} - log_2(\frac{2}{5}) \times \frac{1}{5} = 2.12$$

Por otro lado para calcular la entropía del matching M_1 los valores de $p(x_i)$ se encuentran calculados a continuación en la siguiente tabla:

Par	Frecuencia
$(0,x)$	$\frac{1}{2}$
$(1,x)$	$\frac{1}{2}$

Resultando la entropía del matching como el resultado del siguiente cálculo:

$$I(M_1) = \sum -log_2 p(x_i) = -log_2 \frac{1}{2} \times \frac{1}{2} - log_2 \frac{1}{2} \times \frac{1}{2} = 1$$

A partir de este resultado queda en evidencia que aquellos matchings que tengan la forma de M_1 recibirán una baja puntuación al incorporar la métrica basada en entropía.

Puede verse gráficamente con el ejemplo de la primer figura de la sección 4.1 un matching donde la entropía es mínima. Sin embargo, el cubrimiento del mismo es máximo. Por lo tanto estas dos métricas que intervienen en nuestra definición de evaluación de matching se complementan.

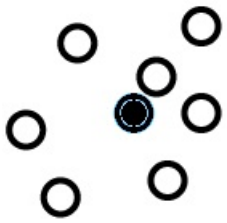


Figura 8: Ejemplo de centroide

2.7. Elección de un representante

Una vez encontrado un matching que satisfaga los criterios descritos(altos valores de correspondencia y entropía), se procede a elegir un representante de cada conjunto. La elección de un representante sintáctico se hace para determinar el cluster sintáctico que le corresponde a una unidad a la hora de predecir su F_0. El representante acústico se determina para poder llevar a cabo la predicción. Para dar con un representante se proponen las siguientes alternativas:

- **Elección al Azar:** En este caso, es simplemente elegir un individuo al azar de cada conjunto. Resulta ventajosa la opción al azar ya que tiene un costo de implementación muy bajo. Pero en contrapartida se pierde certeza de estar eligiendo un elemento representantivo de todo el conjunto.
- **Centroide:** Se denomina centroide al elemento cuya distancia al resto es la menor posible. En este caso se suma complejidad a los cálculos necesarios para poder encontrar dicho elemento; sin embargo se gana en precisión. En la figura 8 puede verse un ejemplo de un centroide de un conjunto, el círculo azul representa el centroide entre los círculos negros.
- **Centroides al azar:** Si suponemos que existen varios elementos cuya distancia a los demás es mínima, entonces se podría elegir al azar dentro del conjunto de individuos que comparten dicha característica, como una variante también puede elegirse al azar dentro del conjunto de elementos que se encuentren en un entorno cercano al centroide.

2.8. Predicción de F_0

El modelo de predicción de F_0 es el que combina todos los aspectos modelados previamente. El modelo de predicción consiste en el proceso por el cual dado un corpus de entrenamiento y mediante métodos no supervisados de clustering se determina el contorno entonacional que debería emplearse en la producción del habla de una oración no perteneciente al corpus. Los pasos del proceso pueden enumerarse de la siguiente forma:

1. Segmentación del corpus en unidades a partir de la transcripción léxica de los enunciados, anotaciones y pitch track de cada enunciado.

2. Análisis y modelado de los aspectos acústicos de las unidades.

3. Análisis y modelado de los aspectos sintácticos de las unidades.

4. Clustering de las unidades en base a sus propiedades acústicas y sintácticas respectivamente utilizando métodos de clustering jerárquico. Este paso es el que caracteriza al modelo de predicción como no supervisado, ya que los métodos de clustering cuentan con esa propiedad.

    * Búsqueda sistemática de una configuración de clustering y pesos vectoriales.

5. Búsqueda de una relación (matching) entre clusters sintácticos y clusters acústicos que sea consistente, con la hipótesis de que existe una correlación positiva entre ambos modelos.

6. Elección de los representantes para el conjunto de clusters acústico y el conjunto de clusters sintácticos.

    * Búsqueda del representante sintáctico (notar que es un vector) que sea más similar a la representación vectorial de la unidad de entrada.

    * Búsqueda del representante del cluster acústico indicado por el matching que relaciona el cluster sintáctico cuyo representante fue encontrado en el paso anterior

7. Predicción de un contorno entonacional a partir del pitch track del representante del cluster acústico indicado.

Nótese que el proceso de predicción se detalla a partir de una unidad, que consiste en la transcripción léxica de una frase sintáctica del enunciado, más sus anotaciones sintácticas y gramaticales. La necesidad de las anotaciones recae en que estas son fundamentales para el fraccionamiento de la oración en unidades o frases sintácticas y su posterior abstracción como fue detallado en el modelo sintáctico. De todas formas si no estuviesen disponibles las anotaciones, existen herramientas automáticas para determinar los POS Tags y la separación en frases sintácticas.

No son de menor importancia los parámetros utilizados durante la fase de clustering, la elección de pesos vectoriales y la elección del representante, ya que de la elección de los mismos depende el rendimiento del sistema de predicción. Hallar una configuración de parámetros es abordado como un problema de optimización y resuelto con una metaheurística greedy. Mediante la metaheurística se obtuvo un conjunto de soluciones sub óptimas, a partir de las cuales se llevaron a cabo los diversos experimentos realizados.

La predicción del sistema es presentada en forma de pitch track, una sucesión normalizada de puntos a lo largo del eje de Tiempo (intervalo [0,1]), en el que cada punto define un valor en Hz que representa la una aproximación a la frecuencia fundamental que deberá emplear un TTS para producir el habla correspondiente a la transcripción léxica del input.

3. Desarrollo

Esta sección tiene como fin describir el desarrollo e implementación del modelo descrito en las secciones anteriores. Durante la etapa de construcción y puesta a prueba del modelo fueron creadas e integradas diferentes componentes de software. También se diseñaron y ejecutaron los experimentos que condujeron a obtener diferentes parámetros del sistema. La construcción del sistema de predicción de F_0 requirió la construcción y desarrollo de sub sistemas menores. El 80 % del sistema fue implementado en Python y el 20 % en Java y Lenguaje Bash. Las etapas del desarrollo pueden resumirse como sigue:

1. Construcción del modelo acústico y desarrollo de las métricas.

   - Evaluación del rendimiento de las métricas usando unidades (frases sintácticas).

2. Construcción del modelo vectorial y desarrollo de las métricas.

   - Evaluación del rendimiento de las métricas usando unidades (frases sintácticas).

   - Construcción del sistema de transformación de información sintáctica en vectores del modelo vectorial.

3. Pre cálculo de todas las distancias acústicas y sintácticas que puedan necesitar los diferentes algoritmos de clustering.

4. Integración con el framework de clustering.

5. Ejecución de las pruebas de los algoritmos de clustering, utilizando valores por defecto. (Los pesos de los features en el modelo vectorial fueron asumidos como 1.)

6. Desarrollo e implementación del algoritmo de matching para los dos conjuntos de clusters.

7. Desarrollo e implementación de la metaheurística elegida para encontrar los parámetros óptimos para la clusterización y matching.

8. Desarrollo del sistema de predicción de F_0.

9. Ejecución de las pruebas y posteriores ajustes del sistema de predicción de F_0.

10. Generación de los resultados del sistema de predicción a partir del conjunto de datos de test perteneciente al corpus.

3.1. Modelo acústico

Durante el desarrollo del trabajo se desarrollaron las medidas de distancia propuestas en la sección 2.3.1 (página 8). En el caso de la distancia acústica se trata de una operación aritmética entre dos pitch tracks. A diferencia de la distancia vectorial, no lleva pesos ni se pondera, de forma tal que puede ser calculada solo una vez, y re utilizar los cálculos durante el resto del

desarrollo. Se construyó un banco de datos con las distancias para cada una de las soluciones propuestas: distancia punto a punto, distancia basada en el grado de correlación y distancia punto a punto mejorada, ver sección 2.3.1 (página 8). El hecho de contar con las distancias ya calculadas mejora el rendimiento del sistema, ya que cada vez que se requiere computar las distancias entre elementos de entrenamiento del corpus para generalizar basta con consultar una base de datos.

3.2. Modelo vectorial

Al desarrollar el modelo vectorial se dejó abierta la posibilidad de ponderar features de los vectores; es decir la capacidad de asignar un peso a cada dimensión. Además se agregó la capacidad de ponderar la componente de distancia vectorial y la componente de distancia de edición. Sin embargo, se desconocía al elaborar el modelo, cuál era el valor adecuado para dichos pesos. Una porción significativa de los experimentos se basó en buscar de forma sistemática dichos valores.

Por ejemplo, se ilustra el efecto de la elección de pesos sobre el clustering: al ponderar las componentes de distancia coseno y distancia de edición se generaron 2 tablas. En la tabla 2 se clusterizó utilizando la distancia coseno solamente con pesos asignados en 1, ponderando con 0 a la distancia de edición. Opuestamente, en la tabla 3 se clusterizó utilizando la distancia de edición solamente, ponderando con 0 a la distancia coseno. Luego, al observar las tablas 2 y 3 queda visible el impacto que tiene en los resultados de los algoritmos de clustering las diferentes valuaciones de peso en la componente vectorial y la componente de distancia de edición.

Cluster	Cardinalidad
0	497(99 %)
1	1 (0.3 %)
2	1 (0.3 %)
3	1 (0.3 %)

Tabla 2: En esta tabla puede verse los efectos que tiene sobre la clustering el uso de pesos para ponderar las diferentes componentes de la distancia del modelo sintáctico. En este caso se anuló la componente de distancia de edición y se asignó pesos iguales a 1 a cada uno de los features del modelo vectorial.

3.2.1. Asignación de pesos en los features

Para el modelo vectorial de los elementos del corpus (Sección 2.4.1), es necesario proporcionar una configuración de pesos para los atributos, de forma tal que se pueda obtener una noción de distancia con la mejor calidad posible; es decir, que dicha función evalúe correctamente elementos que son cercanos. En un primer enfoque se asumió que todos los atributos son igual de relevantes. A medida que se realizaron los sucesivos experimentos se hizo visible que una configuración más efectiva de pesos era necesaria para mejorar la calidad de los resultados.

Cluster	Cardinalidad
0	16 (3 %)
1	3 (1 %)
2	3 (1 %)
3	1 (0.3 %)
4	30 (6 %)
5	440 (88 %)
6	1 (0.3 %)
7	2 (0.6 %)
8	2 (0.6 %)
9	2 (0.6 %)

Tabla 3: En esta tabla puede verse los efectos que tiene sobre el clustering el uso de pesos para ponderar las diferentes componentes de la distancia del modelo sintáctico. En este caso se anuló la componente de distancia coseno.

Los vectores generados a partir del corpus SECYT contienen 17 atributos. Además, es necesario proporcionar un esquema de pesos para la adición entre la componente vectorial y la de distancia de edición, sumando en total 19 los pesos a determinar. Los pesos son números reales dentro del intervalo [0,1]. Dado que el espacio de búsqueda es de dimensiones lo suficientemente extensas como para realizar una búsqueda exhaustiva de la solución exacta, se determinaron empíricamente los pesos mediante el uso de una heurística. En la sección 3.4 se hará foco en la búsqueda de configuraciones de pesos más efectivas que los explicados en la sección actual para obtener buenos resultados.

3.3. Clustering

Como se mencionó en la sección 2.5, se busca mediante mecanismos de clustering agrupar en ambos modelos aquellos elementos del corpus que más próximos se encuentren. A continuación se detalla el proceso de desarrollo e implementación del clustering utilizado y se describen las diferentes alternativas de clustering jerárquica utilizadas. Cabe destacar que las respectivas técnicas no fueron implementadas como parte del trabajo, sino que se utilizó un framework de minería de datos que ofrece la posibilidad de integrar cómodamente las funcionalidades de clustering requeridas.

3.3.1. Integración con Weka

Weka [WF99] es una colección organizada de algoritmos de aprendizaje automático en estado del arte y herramientas para el procesamiento de datos. Las formas básicas de interactuar con los métodos y bibliotecas es mediante la línea de comando o mediante una interfaz de usuario interactiva. Weka permite explorar datos, configurar experimentos de gran escala y realizar cómputo distribuido. Dicho software está implementado en Java y provee múltiples puntos de extensión

ya que su código se distribuye bajo GNU Licencia Publica General.

Utilizando los varios puntos de extensión provistos, se implementó a su vez en Java las modificaciones del sistema versión 3.7.1 para que soporte las medidas de distancia paramétricas desarrolladas para el trabajo aquí presentado. Como parte de la implementación fueron desarrolladas dos herramientas:

- Distancia Coseno & Edición
- Distancia basada en una matriz de datos externa

La metodología en la que los experimentos fueron realizados fue integrando al sistema construido la capacidad de invocación por línea de comando al sistema de clustering utilizando las diferentes opciones que éste brinda. A su vez fue necesario implementar dentro del software Weka la capacidad de recibir paramétricamente los pesos asociados a las diferentes componentes de la distancia vectorial, ya que esta distancia no existía previamente en Weka. Durante el trabajo se realizaron experimentos con todos los algoritmos de clustering jerárquico descriptos a continuación, todos ellos disponibles en el software Weka.

3.3.2. Métodos de clustering aglomerativo

Un procedimiento de clustering jerárquico aglomerativo produce una serie de particiones en los datos $P_n, P_{n-1}, \ldots, P_1$. La primera, P_n consiste en n clusters individuales; la última, P_1, se compone de un solo grupo que contiene los n casos. En cada etapa, el método combina en uno solo, a aquellos dos clusters que estén más próximos uno del otro (es decir los más similares). Las diferentes métodos de agrupamiento surgen de la existencia de múltiples posibilidades a la hora de definir la distancia entre clusters. Estas son las técnicas aglomerativas con las que se experimentó [MRSC08]:

- **Single linkage clustering**: Es uno de los métodos más simples en el clustering aglomerativo jerárquico. La medida de distancia entre clusters que se aplica en este método está definida como la distancia entre el par de objetos más cercanos. Es decir, dado el cluster A y un cluster B, la distancia entre A y B es la mínima distancia entre los pares de individuos i y j con $i \in A$ y $j \in B$. Más formalmente

  $$D(A, B) = Min\{\ d(i, j)\ :\ i \in A\ ,\ j \in B\ \}$$

  En este método se computa la distancia entre cada par de elementos (i, j), donde $i \in A$ y $j \in B$, con A y B clusters de la partición. El valor mínimo de dichas distancias es la distancia entre el cluster A y el cluster B. En otras palabras, la distancia entre dos clusters está dada por el valor del «shortest link» o del enlace más corto. En cada etapa del algoritmo, dados los clusters A y B cuya distancia es mínima, se genera un nuevo cluster C con la unión de A y B.

  Propiedades:

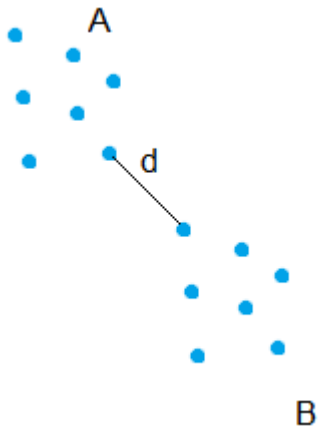


Figura 9: Distancia en Single Link

- Complejidad Temporal: Peor Caso $O(n^3)$
- Complejidad Espacial: $O(n)$

- **Complete linkage clustering**: Define la distancia entre clusters como: Dados A y B clusters,

$$D(A,B) = Max\{\ d(i,j)\ \ : i \in A\ ,\ j \in B\ \}$$

En este caso se toma la distancia entre A y B como la más grande posible entre cada uno de los pares de individuos i y j , con $i \in A$ y $j \in B$. La distancia entre ambos clusters es el tamaño del enlace más largo entre A y B.

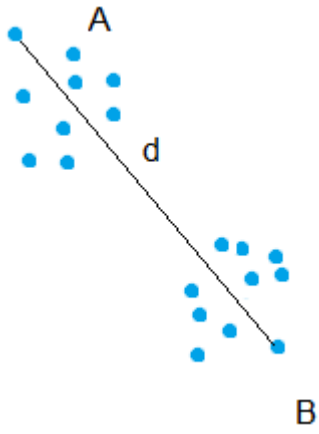


Figura 10: Distancia en Complete Link

Complete-link puede resultar menos conveniente que Single-link ya que se puede dar la siguiente situación: Supongamos que la mejor partición para unir a k antes de haber unido

a j y a i, era necesariamente j o i. Sin embargo luego de unir i con j el mejor candidato para unir con k puede ser un cluster diferente del que resulta de la unión de i con j. En single-link esta situación no es posible. La razón por la cual existe dicha diferencia entre single-link y complete-link es que, en single-link, la distancia definida como la distancia de dos miembros cercanos es una propiedad local que no se ve afectada por la unión de clusters. La distancia que utiliza complete-link no es local al cluster y cambia a medida que cambia el diámetro del mismo.

Propiedades:

- Complejidad Temporal: Peor Caso $O(n^3)$
- Complejidad Espacial: $O(n)$

- **Average linkage clustering**: La distancia entre dos clusters se calcula como el promedio entre todos los pares de objetos

$$D(A, B) = T_{AB}/(N_A \times N_B)$$

Donde T_{AB} es la suma de las distancias para todos los pares (i, j) $i \in A$ $j \in B$ y N_A,N_B son las cardinalidades de los conjuntos A y B respectivamente. En cada iteración del algoritmo, el cluster A y el cluster B tales que $D(A, B)$ es mínima, son combinados en un cluster C. Al aplicar este método, cada cluster se representa por el valor medio de cada variable, esto es, el vector medio. Luego la distancia inter-grupal se define en términos de la distancia de ambos vectores medios.

Propiedades:

- Complejidad Temporal: Peor Caso $O(n^3)$
- Complejidad Espacial: $O(n)$

- **Group-average agglomerative clustering (GAAC)**: El método evalúa la calidad del cluster basado en todas las similaridades entre documentos. A diferencia de los métodos mencionados previamente, donde solo es necesario suministrar la matriz de distancia como entrada, GAAC requiere las siguientes condiciones:

1. Los documentos se representan como vectores.
2. Los vectores tienen norma igual a 1.

Se usa el producto interno como medida de similaridad. La función de distancia calcula la distancia promedio entre todos los pares de documentos, incluyendo pares del mismo cluster.

Propiedades:

- Complejidad Temporal: Peor Caso $O(n^2 \times log(n))$

- Complejidad Espacial: $O(n)$

- **Centroid clustering**: Se denomina centroide al elemento cuya distancia al resto es la menor posible. Su función de similaridad está dada por la similaridad de sus centroides. Propiedades:

  - Complejidad Temporal: Peor Caso $O(n^3)$
  - Complejidad Espacial: $O(n)$

- **Método de Ward**: [War63] Propone un procedimiento de clustering buscando formar particiones $P_n,\ P_{n-1}, \ldots,\ P_1$ de forma tal que se minimice la pérdida de información. En cada paso, la unión entre cada posible par de clusters se evalúa, y se elige el par tal que al unirlo, la pérdida de información resulte mínima. La pérdida de información está definida en términos del criterio de ESS (explained sum of squares). El criterio ESS se define como

$$\mathrm{ESS} = \sum_{i=1}^{n} (\hat{y}_i - \bar{y})^2 .$$

  donde cada y_i representa a un elemento del conjunto a evaluar.

  Propiedades:

  - Complejidad Temporal: Peor Caso $O(n^3)$
  - Complejidad Espacial: $O(n)$

- **Neighbor-joining:** Es un método de clustering bottom-up creado originalmente para encontrar fenogramas (representación en forma de árbol de la similitud biológica de organismos vivos). El método de neighbor-joining[SN+87] comienza con un árbol en el cual existen partes sin completar, incluso la raíz, y con una topología de tipo estrella. Itera hasta que el árbol esté completamente resuelto y todas las longitudes de las ramas sean conocidas.

  Propiedades:

  - Complejidad Temporal: Peor Caso $O(n^3)$
  - Complejidad Espacial: $O(n)$

3.4. En busca de la configuración óptima

Se le da el nombre de configuración al conjunto de pesos para los features instanciados (cada dimensión de la representación vectorial de una unidad y los pesos para la adición entre la componente vectorial y la distancia de edición) y el tipo de enlace para los algoritmos de clustering (Single, Complete, Average, GAAC, Centroid, Ward, o Neighbor-joining) de cada modelo (acústico y sintáctico). Por ejemplo la tabla 4 muestra una posible configuración del sistema de clustering.

Configuración		
Clustering	**Modelo**	**Valor**
	Sintáctico	CENTROID
	Acústico	AVERAGE
Pesos Features Distancia Coseno	**POS Tag**	**Valor**
	A	0.54
	C	0.10
	D	0.88
	E	0.41
	H	0.93
	I	0.46
	J	0.71
	L	0.68
	M	0.22
	N	0.16
	O	0.03
	P	0.61
	S	0.18
	V	0.29
	X	0.81
Pesos Distancia Sintáctica	**Componente**	**Valor**
	Distancia de Edición	0.28
	Vectorial	0.72

Tabla 4: Las primeras 3 filas contienen la configuración de los algoritmos de clustering para el modelo sintáctico y acústico, respectivamente. Luego, se muestra la configuración de los pesos para ponderar los features del modelo vectorial: Cada una de las entradas representa un POS Tag del corpus y con que peso será ponderado. Las últimas 3 filas corresponden a la configuración de los pesos para ponderar la función de distancia del modelo vectorial para las unidades sintácticas

A continuación, buscamos determinar los parámetros de clustering y asignación de pesos tal que la calidad del matching sea máxima. Como calidad óptima se entiende que la relación entre las unidades de ambos modelos tenga atributos que maximicen una función objetivo basada en los siguientes criterios:

- **Cubrimiento del matching**: Se busca que la cantidad de elementos en la relación sea una proporción significativa de los datos de entrenamiento. La cantidad de conjuntos generados para cada modelo puede ser dispar, con lo cual existe la posibilidad de que algunos conjuntos de elementos sean excluidos de la relación, solo se permite que un conjunto este relacionado, con uno y solo uno (Esto se debe a que la predicción es de solo una F_0). Nótese que no son considerados como válidas las relaciones que no cubren en un 100 % a las unidades sintácticas. Como valor experimental se adopto que toda solución debe por lo menos garantizar un cubrimiento del 90 %.
- **Calidad de las relaciones**: Dados dos clusters A y B, se mide la calidad de una relación

entre ambos como:

$$\frac{\#(A \cap B)}{\#(A \cup B)}$$

. Recordemos que el conjunto A es un cluster de unidades clasificado con la métrica definida por el modelo sintáctico y B un cluster de unidades clasificado por la métrica definida por el modelo acústico. Se suma en cada relación el valor de la función de la correspondencia para lograr un único score.

- **Entropía**: La entropía se mide a partir de un matching dado, evaluando la cantidad de información que este provee. Lo que se desea al introducir la componente de entropía en la función objetivo, es evitar que sean buenas soluciones aquellos esquemas de clustering que generan solo un gran conjunto con todos los elementos: Ese tipo de solución es trivial y no aporta información al modelo.

La dimensión del espacio de búsqueda es prohibitivamente grande, está compuesto por 21 componentes, de las cuales 19 son números $\Re_{[0,1]}$ que representan los pesos y los 2 algoritmos de clustering (para el modelo acústico y el sintáctico) que deben ser elegidos entre 6 opciones. Debido a la magnitud del espacio de búsqueda es necesario utilizar un un mecanismo de búsqueda para dar con una solución aproximada para un tiempo de cómputo razonable.

3.4.1. Elección de la metaheurística

Se tuvieron en consideración diferentes las siguientes técnicas heurísticas:

- **Algoritmos evolutivos** [Fra57]: Este método propone explorar el espacio de búsqueda imitando el comportamiento de los organismos biológicos. Durante n generaciones (iteraciones) las soluciones se agrupan en poblaciones y se aplican operaciones de cruzamiento genético y mutación. El algoritmo consiste en, en cada iteración, seleccionar a los mejores individuos de cada población y combinarlos entre sí para ir construyendo a través de varias generaciones soluciones que se aproximen al óptimo. Este método no resultó adecuado ya que no había mucha similitud con los procesos biológicos, ni eran claros los procedimientos de cruzamiento y mutación para las soluciones factibles del sistema. Además, el rendimiento de los métodos evolutivos está fuertemente condicionado por la cantidad de iteraciones, con lo cual la elección del método hubiese insumido más recursos temporales sin una promisoria ventaja por sobre los demás.

- **Colonia de Hormigas** [CDM+91]: Es un método probabilístico para resolver problemas de optimización, está inspirado en el comportamiento que presentan las hormigas para encontrar las trayectorias desde la colonia hasta el alimento. Originalmente creado para buscar un camino óptimo en un grafo con restricciones, opera de la siguiente forma. Inicialmente se deja que las hormigas caminen por el grafo, las hormigas dejan un rastro de feromonas para atraer a más hormigas a dicha solución, con la restricción que las feromonas en el camino caducan después de cierto tiempo. Luego de algunas iteraciones, aquellos

caminos que sean buenas soluciones habrán sido más transitados. Se decidió no utilizar este método ya que inicialmente está orientado a resolver problemas de grafos. Hubiese sido necesario adaptar el problema de encontrar una configuración óptima para el modelo de clasificación a un problema de grafos, que en principio se encontró fuera del alcance del trabajo realizado.

- **Greedy randomized adaptive search procedure** GRASP [FR95]: Un método basado en la selección al azar de una solución inicial dentro de un grupo acotado de soluciones factibles, para luego ir mejorando la solución, usualmente de forma greedy mediante búsqueda local durante n iteraciones. Decidimos utilizar GRASP como heurística en esta tesis debido a que puede interpretarse como buscar un máximo local sobre el gráfico de una función. Por cómo se presentan los criterios para que una solución factible sea buena, fue posible construir una función objetivo basada en dichos criterios. A continuación se presenta información más detallada del método y su aplicación.

3.4.2. Greedy randomized adaptive search procedure

GRASP fue presentado originalmente por Feo y Resende en 1989 [FR95]. Es un método constructivo utilizado típicamente en optimización combinatoria. En un método constructivo un elemento es agregado a una estructura vacía a lo largo de sucesivas iteraciones hasta que la solución del problema es hallada. La elección del ítem a ser incluido en la solución parcial se basa en uno o varios criterios heurísticos que consideran la conveniencia de agregar el ítem a la solución. La metodología depende del problema y también del conocimiento del dominio del problema de quien toma las decisiones a la hora de implementar la metodología. Si una vez agregado un ítem a la solución este vuelve a ser evaluado por las funciones heurísticas se dice que el método es adaptativo.

Además de la función heurística, es necesaria una estrategia de selección entre las soluciones factibles. Una estrategia por defecto suele ser la búsqueda local en forma greedy; es decir, en cada paso en el cual se desea mejorar la solución, elegir aquella que la mejora localmente. Sin embargo, en muchos casos dicha estrategia no tiene una buena performance, sino que es mejor utilizar una estrategia basada en elección al azar dentro de una lista de las mejores soluciones factibles. Esta lista es llamada Restricted Candidate List (RCL). El algoritmo general de GRASP puede verse en el algoritmo 1.

El algoritmo de construcción de la solución greedy con elección al azar sobre la lista de candidatos restringida puede verse en el algoritmo 2. En este caso es importante aclarar que el tamaño de la lista RCL se ve afectada por un coeficiente $\alpha \in (0, 1)$ que determina el balance entre azar y greediness. Es decir, si α es 0 se considera que la heurística es totalmente random, ya que cualquier solución factible puede ser elegida. Si α es 1 entonces la lista RCL solo contiene un elemento y la heurística solo es greedy.

En el caso de las soluciones al sistema de clasificación cualquier asignación de pesos para los POS Tags que esté en el intervalo $[0, 1]$ es posible. En el caso de los algoritmos de clustering

Algoritmo 1 Algoritmo general de GRASP

```
Grasp(Max Iterations,Seed)
  Read Input();
  for k ← 0 → MaxIterations do
    Solution ← Greedy Randomized Construction(Seed);
    Solution ← Local Search(Solution);
    Best Solution ← Mejor Solucion(Solution,Best Solution);
  end for
  return Best Solution;
```

se debe elegir entre todas las posibles opciones detalladas en la sección 3.3.2. Por último falta definir el peso de cada una de las componentes en la distancia sintáctica, en este caso puede ser cualquier número real en el intervalo $[0, 1]$.

Algoritmo 2 Inicialización de la solución

```
Greedy Randomized Construction(Seed)
  Solucion ← ∅
  while Solucion no completada do
    RCL ← Construir Lista(Seed,α);
    Componente ← Elegir Al Azar(RCL);
    Solucion ← Agregar(Componente,Solucion);
  end while
  return Solucion;
```

Por último, el algoritmo 3 es donde se especifica cómo se realizó la búsqueda local, intentando mejorar en cada paso la solución generada en el algoritmo 2. La búsqueda local consiste en evaluar a los candidatos vecinos, considerando vecinos a aquellas soluciones que se encuentran próximas. Se considera próxima una solución que puede alcanzarse a través en un desplazamiento de $\beta \in [0, 1]$ sobre cada peso de los features y cada peso en distancia sintáctica. Es importante destacar que el algoritmo de búsqueda local no realiza todos los incrementos/decrementos en los pesos al mismo tiempo, sino que busca progresivamente una dirección de crecimiento en cada peso individualmente. Una vez que las direcciones de crecimiento están determinadas se prosigue a avanzar en cada una de ellas hasta encontrar un máximo local. El coeficiente de incremento β es adaptativo, eso quiere decir que si en la búsqueda de un máximo local la solución pierde calidad respecto a la anterior, se continua incrementando con una reducción de un orden de magnitud de β.

3.4.3. Elección de los parámetros

La aplicación de la heurística se encuentra condicionada por la cantidad de iteraciones y dos coeficientes, α y β.

- **Cantidad de Iteraciones:** La cantidad de iteraciones con las cuales se ha experimentado el procedimiento del algoritmo 1 fueron 100, 300, 500, 1000, 2000 y 3000 iteraciones.

Algoritmo 3 Mejora de la solución

```
Local Search(Solution)
  Mejor Solution ← None
  for k ← 0 →Dimensiones de búsqueda do
    if EsDimensionCrecimiento(k) then
      Solution ← BusquedaLocalKEsimaDimension(Solution,k);
      Best Solution ← MejorSolucion(Solution,Best Solution);
    end if
  end for
  return Best Solution;
```

- **Coeficiente de RCL:** Como se mencionó, es necesario contar con un número real $\alpha \in [0, 1]$ que sirve como parámetro para generar la lista RCL. El parámetro α brinda el flexibilidad entre greediness y aleatoriedad en la heurística. Empíricamente se eligió $\alpha = \frac{1}{2}$. De esta forma la elección sobre la lista RCL tiene un balance adecuado entre su componente de azar y de greediness.

- **Coeficiente de avance:** El coeficiente β se utiliza como medida de avance para desplazarse entre soluciones vecinas al realizar la búsqueda local. La búsqueda local fue desarrollada para que se adapte a la posibilidad de que en incrementos de β la solución pase por un máximo local sin efectivamente encontrarlo. En la práctica se trabajó con $\beta = \frac{1}{5}$ y $\beta = \frac{1}{100}$ en caso de que una solución vecina sea peor que la que se está visitando. De esa forma se intenta no encerrar un máximo local entre dos soluciones vecinas.

3.4.4. Detalles de implementación

Esta sección tiene el fin de reportar el trabajo realizado en el desarrollo y aplicación de la metaheurística. Debido a la gran cantidad de soluciones a explorar para acercarse a la configuración asociada a una solución óptima o suficientemente buena, fue necesario aprovechar al máximo el tiempo de cómputo y los recursos facilitados para la ejecución de la tesis por el Departamento de Computación. La implementación de GRASP se escribió en lenguaje Python 2.5. Además de la metaheurística se implementó un motor de ejecución paralela para explotar el poder de cómputo en los momentos en los cuales los laboratorios del Departamento de Computación se encontraban ociosos. En total fueron requeridas alrededor de 360 horas de cómputo en un total de 36 computadoras (Procesador: Intel(R) Core(TM)2 Quad CPU Q9550 @ 2.83GHz, Memoria: 3GB). Dando como resultado casi 13.000 horas de cómputo. La fase de clusterización fue la que más tiempo insumió. Un cuello de botella fue el uso del file system distribuido sobre el cual operan los recursos de los laboratorios, ya que para evitar el recálculo de las matrices de distancia y los clusters acústicos, estos datos fueron persistidos en archivos.

3.5. Sistema de predicción de F_0

El algoritmo 4 presenta el método de predicción de F_0 desarrollado. En él se hace referencia a los módulos del sistema detallados previamente. El algoritmo recibe como parámetros: la representación vectorial de una unidad, una configuración (un ejemplo de la configuración puede verse en la sección 3.4, página 31) y un criterio para la elección del representante (centroide, centroide al azar y azar, vistos en detalle en la sección 2.7, página 23).

En los primeros 3 pasos se obtienen los clusters acústicos y sintácticos, así como el matching resultante para la configuración provista. Luego se procede a obtener los representantes sintácticos, uno para cada cluster sintáctico. A continuación, en la línea 4, invocando a la función *determinar_cluster_sintactico* cuyos parámetros son los representantes sintácticos, el vector de entrada y la configuración, se determina cuál es cluster cuyo representante es más similar al vector de entrada. Una vez determinado el cluster que le corresponde al vector de entrada, se consulta el matching, para saber cuál es el cluster acústico correspondiente. El representante del cluster acústico hallado representa el output del algoritmo.

Es importante destacar que el criterio de selección de representante puede impactar fuertemente en la variabilidad de los resultados. Si se utiliza el criterio del centroide los representantes no varían entre sucesivas inputs, lo cual no ocurre si se elige el criterio de selección de representante al azar.

Algoritmo 4 Algoritmo de predicción de F_0

generalizar(input_vector,configuracion,criterio):

1: clusters_acusticos ← clusterizar_modelo_acustico(configuracion)
2: clusters_sintacticos ← clusterizar_modelo_sintactco(configuracion)
3: matching ← generar_matching(clusters_acusticos,clusters_sintacticos)
4: cluster_sintactico ←determinar_cluster_sintactico(clusters_sintacticos,input_vector, configuracion)
5: cluster_acustico ← matching[cluster_sintactico]
6: representante_acustico ← obtener_representante_acustico(clusters_acustico,criterio)
7: **return** obtener_pitch_track(representante_acustico)

4. Resultados

Presentamos primero en la sección 4.1 los resultados obtenidos durante la construcción de las diferentes partes del sistema de predicción de F_0: el clustering sintáctico, el clustering acústico y el matching conectando a ambos. En otras palabras, presentamos las configuraciones encontradas mediante el empleo de la metaheurística GRASP.

Luego la sección 4.2 muestra los resultados de la evaluación del sistema de predicción de F_0. Se comparó en cada caso de test la calidad de la predicción obtenida con la calidad la predicción utilizando dos baselines. La calidad de una predicción se calculó como la distancia acústica entre el pitch track predicho y el pitch track del gold standard.

4.1. Clustering

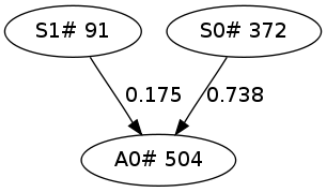
Los resultados del clustering son presentados a través de un muestreo representativo de los resultados obtenidos: no exitosos, promedio y la mejor configuración encontrada. Se presenta también la performance de la metaheurística en función de la función objetivo. Por último, estudiamos la correlación que existe entre el modelo acústico y el modelo sintáctico, utilizando la noción de distancia como métrica.

4.1.1. Soluciones representativas de las configuraciones obtenidas

Los gráficos a continuación presentan algunos de los resultados de clustering de las unidades del corpus, junto al mejor matching para el esquema de clasificación obtenido. Debido a la explosión combinatoria que se genera al explorar todas las posibilidades de parámetros con los que se ha experimentado, se han seleccionado aquellos experimentos que fueron más representativos de todo el conjunto de pruebas realizadas. Los resultados presentados a continuación corresponden a las 10 mejores soluciones encontradas por GRASP explicado en la sección 3.4.2 (página 34). Los resultados son presentados en orden creciente en función la calidad de las soluciones encontradas. La interpretación de los gráficos de clasificación es la siguiente, en la parte superior se encuentran los conjuntos sintácticos rotulados con una letra S mayúscula, en la parte inferior se encuentran los conjuntos acústicos, rotulados con una letra A. El número que acompaña el rótulo, corresponde a la cardinalidad del conjunto. Las flechas que están trazadas desde los conjuntos sintácticos hacia los acústicos representan el matching y el número que compaña la flecha indica el valor de la función de evaluación del matching para cada conjunto. Resultando el valor total de la función de evaluación del matching la suma de cada una de las relaciones.

A continuación puede verse un representante de una configuración que derivó en un resultado no exitoso. Recordemos que se consideran no exitosos a los resultados de clustering que no cubren una porción significativa de los datos, o que caen en casos triviales de clusterización. A izquierda se ve la tabla que describe la configuración y a la derecha la representación gráfica del clustering y el matching.

Link Type Modelo Sintáctico	Complete
Clusters modelo Sintáctico	2
Link Type Modelo Acústico	N. Joining
Clusters modelo Acústico	1
Cubrimiento	91 %
Función objetivo	$\frac{55}{100}$



A continuación se presenta una configuración de clustering que representa a las soluciones promedio. Las soluciones que son calificadas como promedio, son aquellas que por ejemplo si bien no caen un caso trivial de clustering, agrupan de forma asimétrica a las unidades. La tabla 5 junto a la figura 11 presentan la configuración y su representación gráfica respectivamente. En esta configuración la entropía es mejor que en el caso trivial, sin embargo la métrica basada en maximizar la función objetivo, (página 21) tiene en promedio valores muy bajos ya que el clustering sintáctica tiene un conjunto que concentra a más de la mitad de los individuos, esto lleva a tener un cubrimiento del 68 % de las unidades.

Link Type Modelo Sintáctico	Average
Clusters modelo Sintáctico	9
Link Type Modelo Acústico	Average
Clusters modelo Acústico	6
Cubrimiento	68 %
Función objetivo	$\frac{65}{100}$ en escala [0,1]

Tabla 5: Configuración de clustering de una solución promedio

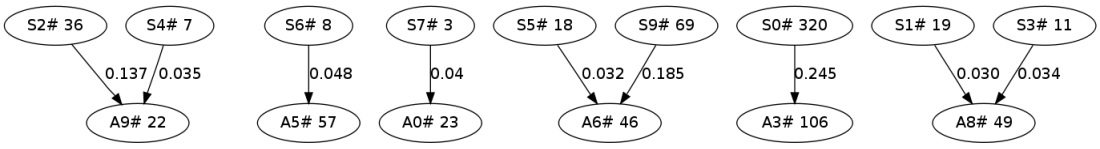


Figura 11: Matching utilizando un modelo(AVERAGE, AVERAGE)

A continuación se presenta la mejor solución encontrada por la metaheurística, esta fue la configuración elegida para el resto de los experimentos de esta tesis.

Link Type Modelo Sintáctico	Centroid
Clusters modelo Sintáctico	9
Link Type Modelo Acústico	Average
Clusters modelo Acústico	6
Cubrimiento	98 %
Función objetivo	$\frac{70}{100}$ en escala [0:1]

Tabla 6: Configuración de clustering de la mejor configuración encontrada por GRASP

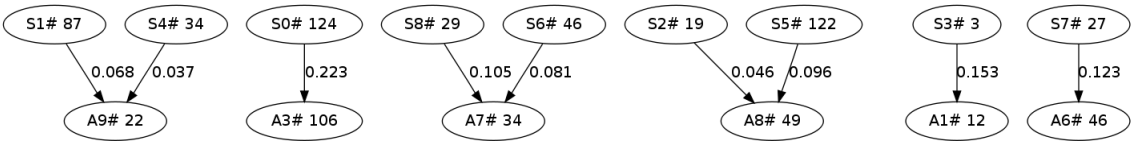


Figura 12: Matching utilizando un modelo (CENTROID,AVERAGE) en el cual cada uno de los clusters sintácticos está relacionado con solo un cluster acústico y el grado de incidencia de cada cluster acústico es bajo (máximo 2), por otro lado se da un alto cubrimiento de las unidades. Dichas condiciones son propensas para que la función objetivo le otorgue un puntaje favorable.

4.1.2. Performance de la metaheurística

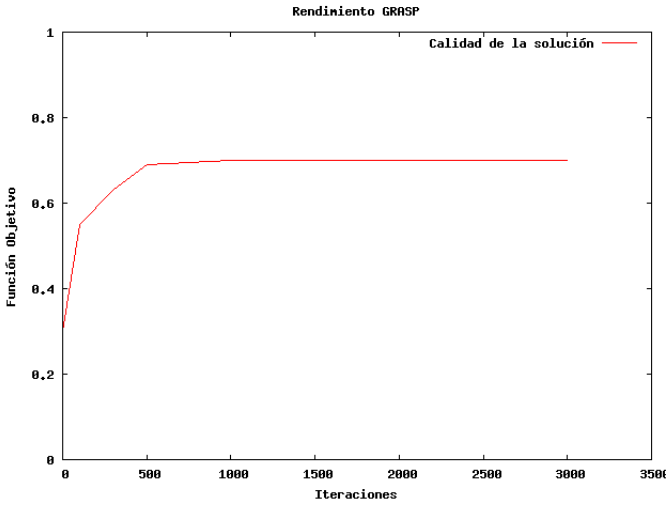


Figura 13: Performance de la metaheurística de GRASP

La figura 13 muestra el valor de la función objetivo (cuán buena es la relación encontrada entre los clusters sintácticos y los acústicos) en función de la cantidad de iteraciones. Observamos

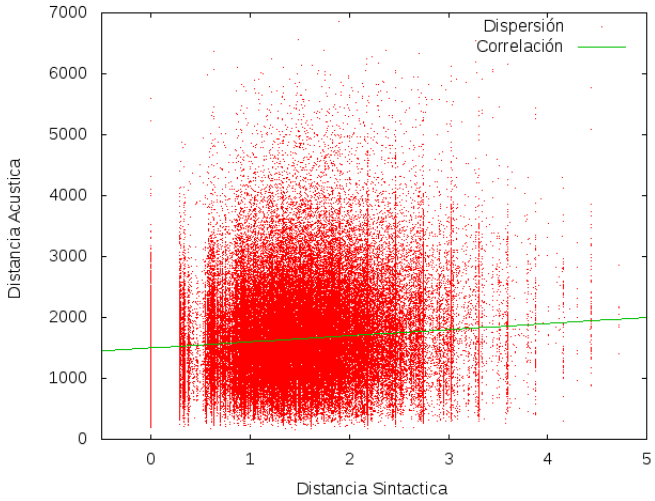


Figura 14: Correlación lineal entre la distancia acústica y la distancia sintáctica para las unidades del corpus

que el valor de la función objetivo se mantiene estable a partir de las 1000 iteraciones, razón por la cual se decidió no aumentar el número de iteraciones más allá de ese punto. El gráfico corresponde a la ejecución de la metaheurística con parámetro $\alpha = \frac{1}{2}$; se experimentó con otros valores de $\alpha = \frac{1}{2}$, y los resultados obtenidos fueron similares a los presentados en la figura 13.

4.1.3. Correlación de los modelos

Una de las hipótesis centrales del trabajo realizado es que existe una correlación positiva entre la medida de distancia del modelo acústico y la medida de distancia del modelo sintáctico. Es decir, ¿dos unidades que son similares sintácticamente lo son también acústicamente?, y viceversa. Para validar dicha hipótesis se estudió si existe una correlación entre todo par de unidades de los datos de entrenamiento, de acuerdo a las definiciones de distancia elegidas. Los resultados del test de correlación de Pearson's indican que hay una correlación moderada, pero positiva y significativa, como puede verse en la figura 14 (Pearson: $\rho = 0.15$; t-test: p-valor$< 10^{-16}$).

4.2. Sistema de predicción de F_0

En esta sección se presenta la evaluación del sistema con los siguientes resultados:

- Un muestreo general de las predicciones obtenidas.
- La performance del sistema de predicción como función del criterio de elección del representante.
- La variación en la performance del sistema de predicción en función de la cantidad de datos de entrenamiento (lo que llamamos *escalabilidad* del sistema).

4.2.1. Baselines y Gold standard

Los baselines con los que se evaluó la calidad de las predicciones obtenidas por nuestro de sistema de predicción de F_0 fueron desarrollados a partir de los mismos datos de entrenamiento. En un primer enfoque se compararon los resultados contra un baseline constante y en una segunda etapa se hicieron comparaciones contra un baseline basado en un modelo de ajuste lineal sobre las unidades de entrenamiento:

- Baseline Constante: El primer baseline para contrastar los resultados se basa en un promedio del pitch track de todos los elementos del corpus de entrenamiento.
- Baseline Lineal: El segundo baseline consiste en ajustar un modelo lineal a cada pitch track y luego obtener una ordenada al origen y una pendiente promedio. La elección de este baseline está basada en una motivación fonológica: Habitualmente las oraciones declarativas poseen una entonación descendente y no constante. Por lo tanto, modelar esos contornos entonacionales con una recta descendente es más adecuado que con una constante.

Para la evaluación de resultados se tuvo en cuenta la similitud entre el pitch track de la unidad de test y una predicción del sistema. De esta forma, evaluamos el sistema de predicción de F_0, el baseline constante y el baseline lineal.

La evaluación se realizo sobre los datos de test. Las pruebas con el sistema de predicción de F_0 se realizaron utilizando como input las transcripciones léxicas de las unidades de test y comparando el pitch track predicho por el sistema de predicción de F_0 con el pitch track original de la unidad.

Es importante destacar que el gold standard usado en nuestra evaluación consiste en contornos entonacionales producidos por una hablante puntual. Es decir, nuestro gold standard representa sólo una de las muchas posibles formas de producir las oraciones del corpus. Una consecuencia de esta elección es que algunas predicciones del sistema que sonarían bien a un oído humano podrían recibir una baja puntuación, debido a que difieren del gold standard correspondiente.

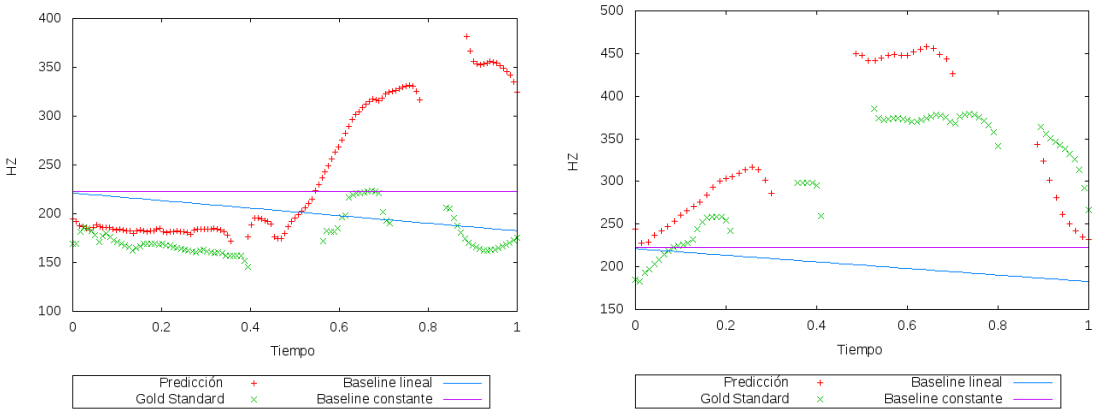
Una forma ideal de evaluar la naturalidad de las predicciones realizadas por el sistema, independiente de un gold standard, sería mediante tests perceptuales. En tales tests, se generarían estímulos usando un sistema TTS que asigne F_0 usando las predicciones de nuestro sistema; tales estímulos serían luego mostrados a humanos para su evaluación subjetiva de la naturalidad. Otra manera de evaluar los resultados podría ser grabar a varios hablantes distintos produciendo las oraciones de los datos de test. Entonces, cada predicción del sistema se evaluaría de acuerdo a su cercanía a uno o más de los contornos entonacionales producidos por los hablantes. Se deja estos métodos alternativos de evaluación de nuestro sistema de predicción de F_0 como posibles direcciones de trabajo futuro.

4.2.2. Resultados representativos de la performance del sistema

Podemos agrupar los resultados en tres categorías: exitosos, no exitosos y promedio. Los resultados exitosos son aquellos en los que las predicciones superan a los baselines en calidad,

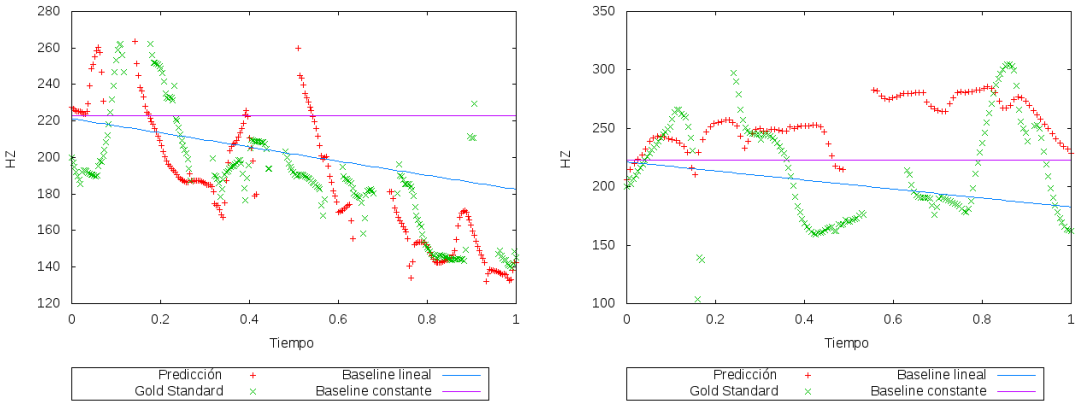
y además se aprecia visualmente que el pitch track predicho es muy similar al pitch track del gold standard. Los resultados no exitosos son aquellos en que las predicciones son peores que los baselines, y además se aprecia visualmente que el pitch track predicho no se parece al pitch track del gold standard. Por último los resultados promedio son aquellos en que las predicciones son numéricamente peores que los baselines aunque visualmente se aprecie una marcada similitud.

Se presenta un muestreo gráfico de los resultados obtenidos discriminando por la calidad de los resultados. En cada caso se eligieron 2 predicciones para ilustrar el desempeño del sistema de predicción. La figura 15 muestra dos casos los cuales las del sistema superan las predicciones de ambos baselines. La figura 16 presenta dos casos donde las predicciones del sistema son peores que las predicciones de alguno de ambos baselines, aunque al observar el contorno entonacional los resultados son mejores de lo que aparentan numéricamente. Por último, la figura 17 presenta dos casos en donde las predicciones obtenidas no son favorables para la asignación de un pitch track, por lo que ambos baselines, tienen un mejor desempeño.



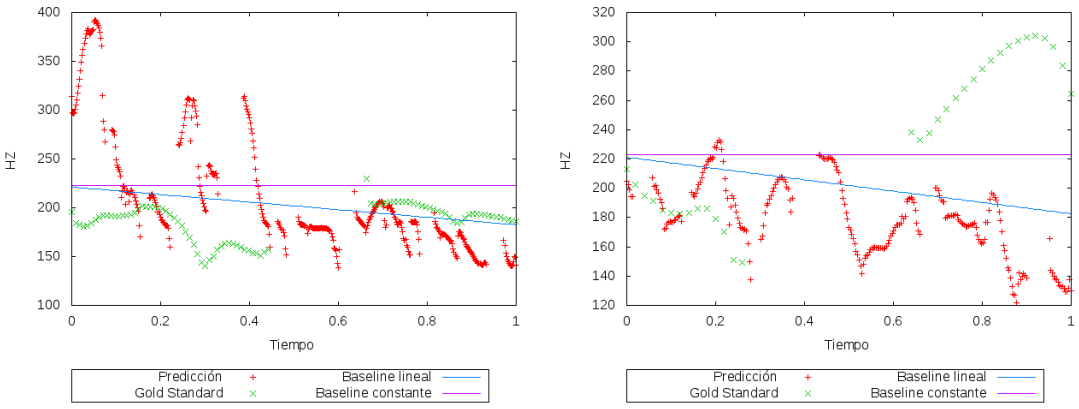
(a) (b)

Figura 15: Dos ejemplos en los cuales la predicción del sistema es mejor que la aproximación mediante ambos baselines. En el caso de 15(a) el pitch track predicho corresponde «Los rebeldes» y el pitch track del gold standard corresponde a «Los hombres del islam». En el caso de 15(b) el pitch track predicho corresponde a «Un hundimiento» y el pitch track del gold standard corresponde a «El manuscrito»



(a) (b)

Figura 16: Dos ejemplos en los cuales las predicciones del sistema son peores que las predicciones de alguno de ambos baselines, aunque al observar el contorno entonacional los resultados son mejores de lo que aparentan numéricamente. En el caso de 16(a) el pitch track predicho corresponde a «El dermatólogo» y el pitch track del gold standard corresponde a «Los camiones cargados». En el caso de 16(b) el pitch track predicho corresponde a «Es inmodesto y servil» y el pitch track del gold standard corresponde a «El ágil acróbata intrépido».



(a) (b)

Figura 17: Dos ejemplos en los cuales la predicción del sistema es peor que la aproximación mediante ambos baselines. En el caso de 17(a) el pitch track predicho corresponde a «Cientos de edificios» y el pitch track del gold standard corresponde a «Una rústica mampara». En el caso de 17(b) el pitch track predicho corresponde a «Sobran» y el pitch track del gold standard corresponde a «El manuscrito»

4.2.3. Evaluación numérica de los sistemas de predicción

En las tablas : 7, 8 y 9 puede observarse la performance de las 10 mejores configuraciones para los diferentes criterios de selección del representante: al azar, centroide y centroide al azar. Cada tabla está ordenada de menor a mayor en función de la cantidad de predicciones del sistema que superaron a ambos baselines y a cada baseline por separado.

Mejor que ambos baselines	Mejor que baseline constante	Mejor que baseline lineal
9.75 %	9.75 %	13.0 %
7.5 %	8.0 %	11.2 %
6.25 %	8.0 %	9.25 %
4.5 %	4.5 %	7.5 %
4.25 %	4.25 %	7.25 %
4.0 %	4.0 %	5.25 %
3.75 %	3.75 %	6.0 %
3.5 %	3.5 %	5.0 %
3.0 %	3.5 %	4.25 %
2.0 %	2.0 %	3.5 %

Tabla 7: Performance del sistema de predicción de F_0 basando la elección del representante al azar para las 10 mejores configuraciones obtenidas durante la fase de clustering

Mejor que ambos baselines	Mejor que baseline constante	Mejor que baseline lineal
8.0 %	8.0 %	11.2 %
2.75 %	2.75 %	13.0 %
2.25 %	8.0 %	2.25 %
3.5 %	3.5 %	7.5 %
3.25 %	3.25 %	7.25 %
3.0 %	3.0 %	5.25 %
3.75 %	3.75 %	6.0 %
3.5 %	3.5 %	5.0 %
3.0 %	3.5 %	3.25 %
2.0 %	2.0 %	3.5

Tabla 8: Performance del sistema de predicción de F_0 basando la elección del representante con el criterio del centroide para las 10 mejores configuraciones obtenidas durante la fase de clustering

Mejor que ambos baselines	Mejor que baseline constante	Mejor que baseline lineal
12.5 %	12.7 %	17.0 %
12.2 %	12.7 %	18.5 %
11.2 %	12.0 %	16.2 %
9.25 %	9.75 %	12.2 %
9.0 %	9.0 %	15.2 %
8.0 %	8.0 %	12.2 %
7.25 %	7.5 %	10.0 %
4.75 %	5.25 %	6.5 %
4.5 %	4.5 %	5.5 %
2.25 %	2.5 %	4.0 %

Tabla 9: Performance del sistema de predicción de F_0 basando la elección del representante con el criterio de centroide al azar para las 10 mejores configuraciones obtenidas durante la fase de clustering

El método trivial de elección de representante (selección al azar) superó en performance al método standard para clustering (centroide). Esto sucede porque cuando se utiliza el criterio del centroide se está fijando a los representantes para todos los casos de prueba, intentando predecir todo el conjunto de datos de prueba con solo una diminuta porción de los datos de entrenamiento. Cuando se utiliza el criterio al azar se están prediciendo los datos de test con todos los datos de entrenamiento, sin embargo se corre el riesgo de elegir unidades poco representativas de cada cluster como representantes. A partir de este resultado se combinaron ambos criterios dando lugar a un método híbrido y se logró la mejor performance. La metodología para asignar un representante en el sistema de predicción de F_0 es seleccionar el centroide al azar de los clusters acústicos.

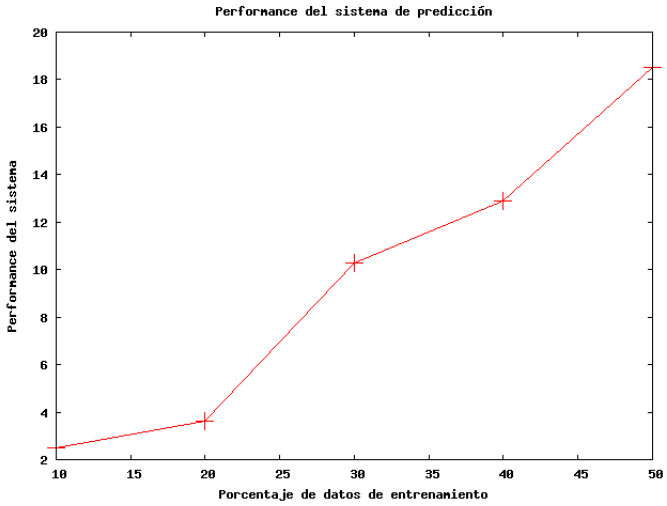


Figura 18: Performance del sistema de predicción en función de la cantidad de datos de entrenamiento

4.2.4. Escalabilidad

Como el corpus con el que se trabajó es muy pequeño (recordemos que cuenta en total con solo 741 enunciados de los cuales la mitad se utilizó para entrenamiento y se segmentó en 500 unidades) se realizó una simulación para determinar el rendimiento del sistema de predicción en caso de disponer de más datos de entrenamiento. En la figura 18 puede verse la progresión de la performance en la simulación.

En la siguiente tabla se presenta la performance del sistema de entrenamiento a medida que se varían el porcentaje de unidades del corpus utilizadas durante la fase de entrenamiento. Para determinar la escalabilidad del sistema se decidió mantener constante el conjunto de datos de test utilizados para medir la performance, con el fin de evitar variaciones causadas en caso de que los datos de test no sean homogéneos.

Datos de entrenamiento	Datos de test	Performance
10 %	50 %	2.51 %
20 %	50 %	3.64 %
30 %	50 %	10.27 %
40 %	50 %	12.9 %
50 %	50 %	18.5 %

Tabla 10: Progresión de la performance del sistema de predicción en función de la cantidad de datos de entrenamiento utilizados manteniendo los datos de test.

5. Conclusiones y Trabajo futuro

En la sección 5.1 presentamos la discusión de los resultados presentados en la sección 4 y además presentamos las conclusiones de la tesis y el cumplimiento de los objetivos propuestos. Por último en la sección 5.2 planteamos las posibles mejoras a realizar en un futuro.

5.1. Conclusiones

A continuación se presenta la discusión y conclusiones del trabajo a partir de los resultados presentados en la sección 4. Se discutirá primero los resultados de clustering (ver sección 4.1). Por último los resultados del sistema de predicción (ver sección 4.2).

5.1.1. Clustering

Los resultados exhibidos en la sección 4.1 presentan la performance de los distintas estrategias detalladas en las sección 3.3.2. Se observó que la estrategia Single Link representa un caso patológico, solo es eficiente para clusterizar cuando las distancias entre los elementos son grandes. En el marco de la tesis, las distancias entre las unidades están distribuidas dentro de un rango, sin grandes saltos intermedios, de acuerdo a como indica la figura 20. Por lo tanto el algoritmo de Simple Link resultó inadecuado para este trabajo. Por otro lado, se observó que la mayoría de las configuraciones que logran que el sistema de predicción se desempeñe con performance corresponden a la estrategia de clustering WARD para el modelo sintáctico y AVERAGE para el modelo sintáctico. Otra estrategia exitosa fue CENTROID para el modelo acústico y MEAN para el modelo sintáctico.[1]

5.1.2. Metaheurística

De acuerdo a lo observado en la figura 13 (página 40) el algoritmo de búsqueda alcanza un máximo de la función objetivo luego de las 1000 iteraciones (el máximo obtenido fue de un 70 % del puntaje posible). El comportamiento se mantuvo constante al variar los parámetros α y β,[2] y la cantidad de iteraciones. Tal comportamiento de la metaheurística puede deberse a que la función objetivo tenga un sesgo. Durante la búsqueda sistemática se comprobó que aumentar el grado de azar en la lista de candidatos factibles no produce mejores resultados que si se itera con una lista altamente restrictiva. Sin embargo al aumentar la aleatoriedad de la lista de candidatos, la cantidad de tiempo de cómputo requerido para finalizar cada iteración se tornaba prohibitiva.

[1] En ambos casos cuando se refiere a las configuraciones, se hace alusión tomando las estrategias de a pares. (WARD-AVERAGE y CENTROID-AVERAGE)

[2] Recordemos que los parámetros α, β indican el grado de azar al construir la lista RCL y el tamaño del desplazamiento al buscar soluciones vecinas respectivamente

5.1.3. Predicción de F_0

En el presente trabajo se planteó como hipótesis la existencia de una relación entre los aspectos sintácticos y acústicos de una oración. En base a esto se desarrolló un modelo acústico y un modelo sintáctico, a partir de los cuales se extrajeron definiciones de distancia (acústica y sintáctica) que condujeron a una serie de experimentos de clusterización. A partir de los resultados exitosos de los experimentos de clusterización se desarrolló un sistema que tiene la capacidad de predecir la frecuencia fundamental que un sistema de TTS deberá utilizar para sintetizar una frase dada. Al analizar los resultados de la sección 4.1.3 se consiguió verificar la hipótesis central del trabajo.

Este trabajo significa un cambio de enfoque ya que los sistemas actuales de asignación de prosodia se basan en modelos supervisados. En estos sistemas las anotaciones cumplen un rol clave y la problemática se da porque esta información debe ser generada por personas calificadas, una actividad costosa en términos temporales y de recursos humanos.

Luego de haber realizado los experimentos de predicción de F_0, se observó que el rendimiento del sistema es: muy bueno en algunos casos (20 %), promedio en la mayoría de los casos y presenta una cola de resultados negativos. Se ha observado que las predicciones obtenidas de mejor calidad son las basadas en centroide al azar, seguidas por las obtenidas con el criterio de representante al azar y por último aquellas que se generan utilizando el criterio de centroide. Durante el desarrollo del modelo acústico y del modelo sintáctico se pensó que la mejor predicción sería aquella basada en centroide. Sin embargo al contrastar la idea con los resultados obtenidos se encontró muy bajo su rendimiento. El bajo rendimiento se debe a que que solamente se utiliza un conjunto acotado y estático por configuración para realizar las predicciones (ya que los centroides se calculan una vez para cada configuración de clustering) para predecir todo el cuerpo de datos de test. Si se tiene en cuenta que los datos de entrenamiento no son lo suficientemente homogéneos, puede verse que con un conjunto estático y acotado de unidades no puede predecirse todos los datos de test adecuadamente. Sin embargo, la idea de utilizar un centroide al azar (en la mayoría de los casos se tomó un centroide y su entorno más cercano) resultó mejor de lo esperado, ya que la selección al azar dentro de un cluster resultó más representativa.

Los resultados de escalabilidad (sección 4.2.4) indican que, si se produce un aumento del tamaño del corpus, los resultados tienden a mejorar. Este resultado permitirá en un futuro re entrenar al sistema con colecciones masivas de datos (como por ejemplo YouTube o colecciones de audio en Internet) para lograr resultados de alta calidad, y así construir sistemas mejorados de TTS.

En muchas predicciones cuya calidad se ubica por abajo de las predicciones baseline puede apreciarse visualmente que el contorno entonacional de la predicción es en realidad bastante similar al gold standard. Esto sugiere que a futuro deberán perfeccionarse las definiciones de distancias utilizadas, ya que son de gran utilidad para resolver otros problemas del dominio o extender este trabajo.

5.2. Trabajo futuro

Podemos identificar ciertos puntos del trabajo que pueden ser mejorados en un futuro o presentan puntos de extensión para resolver problemas relacionados. Se presentan las mejoras planteadas por cada módulo de la tesis.

Modelo acústico Mejorar la distancia acústica, para que pueda tener en cuenta mediciones faltantes. Esto permite que si existen pitch tracks con algún gap, la medida de distancia sea tolerante a este tipo de errores. Sería positivo incorporar algún tipo de detección de ruido y outliers en el pitch track por medio de filtros y técnicas estadísticas, para darle al sistema más robustez y la capacidad de adaptarse a diferentes cuerpos de datos.

Modelo sintáctico Una alternativa a considerar es extender el modelo vectorial para trabajar con oraciones segmentadas a un nivel más profundo del árbol sintáctico. Se espera que esta nueva granularidad junto a la mejora propuesta en el modelo acústico logre reforzar la evidencia de la existencia de la correlación entre el modelo acústico y el modelo sintáctico.

Metaheurística Como se mencionó en la sección 5.1.2 (página 48), la metaheurística encuentra un tope para la función objetivo, a pesar de variar los parámetros. Como trabajo futuro surge la necesidad de mejorar la lista de candidatos factibles.

Clustering Existen modificaciones que pueden realizarse sobre la implementación de los algoritmos para optimizar el uso de recursos y acelerar su ejecución. Estas modificaciones consisten en implementar los algoritmos en algún lenguaje de bajo nivel, como C++. Como mejora podría estudiarse el desempeño de otros métodos no supervisados para clasificación automática.

Corrimiento lineal Según lo discutido en la sección 5.1.3 algunos resultados del sistema no logran reflejar numéricamente su buena calidad, aunque al observar los gráficos los contornos son adecuados. En este caso es necesario identificar los casos en los que un *corrimiento lineal* mejore las calidad de las predicciones del sistema. En la figura 19 se muestra un ejemplo en que es necesario un corrimiento sobre el eje y.

Predicción Caracterizar a las unidades de entrada que reciben predicciones acertadas podría aumentar significativamente la performance general del sistema. De este modo, se podría evitar utilizar el sistema para aquellas entradas para las cuales ya se sabe antemano que las predicciones no serán adecuadas. Por último, podrían estudiarse nuevas alternativas para generar F_0 a partir de los pitch tracks de varios representantes acústicos, en lugar de usar el pitch track de un solo representante.

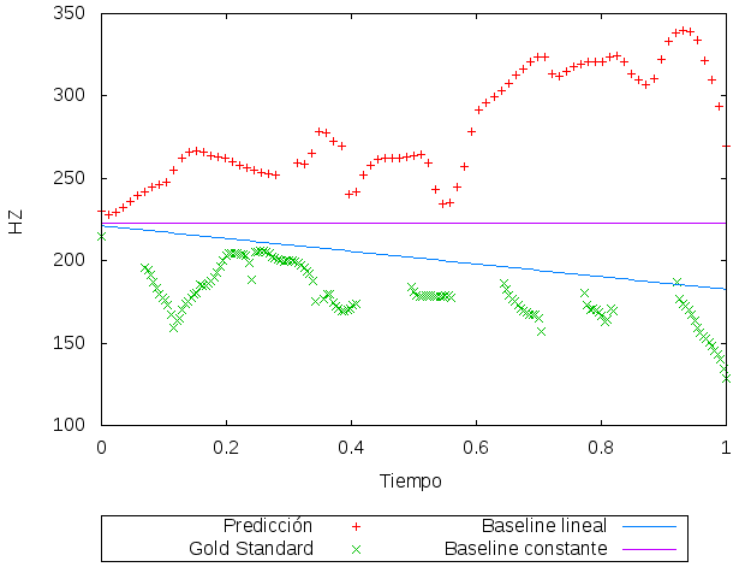


Figura 19: En la siguente figura se puede observar como un corrimiento lineal sobre la predicción causaría una disminución en la distancia entre ambos contornos entonacionales

A. Valores Experimentales

Esta sección presenta o justifica valores, constantes u otros parámetros que fueron obtenidos de forma experimental para hacer posible o mejorar el rendimiento de diversos componentes tanto del modelo como de la implementación así como de la estructura del corpus

A.1. Modelo Acústico

A.1.1. Similitud entre oraciones VS Similitud entre unidades

Durante el experimento fue puesta a prueba la función de similitud acústica basada en intervalos. En los histogramas (ver figuras 20 y 21) se grafica el porcentaje de elementos del corpus en función de la distancia entre dichos elementos. En particular puede verse en la figura 20 que más de la mitad de los unidades (frases sintácticas) están a una distancia menor a 0.25. Sin embargo en la figura 20 se ve que las distancias entre las oraciones completas está concentrada al rededor del 0.375

En ciertos casos patológicos se manifiesta una propiedad no deseada que posee la función de similitud, indica que dos unidades son cercanas incluso si existe algún intervalo con pronunciadas diferencias, para ejemplificar mejor este caso es conveniente ver la figura 23 donde existe una gran similitud sobre la cola de la función pero no al comienzo. Sin embargo, existen casos en los que se comporta adecuadamente como en la figura 22.

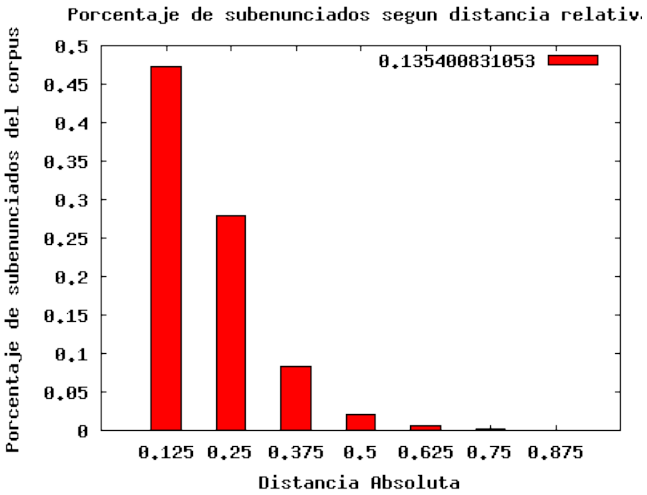


Figura 20: Histograma distancias relativas entre unidades

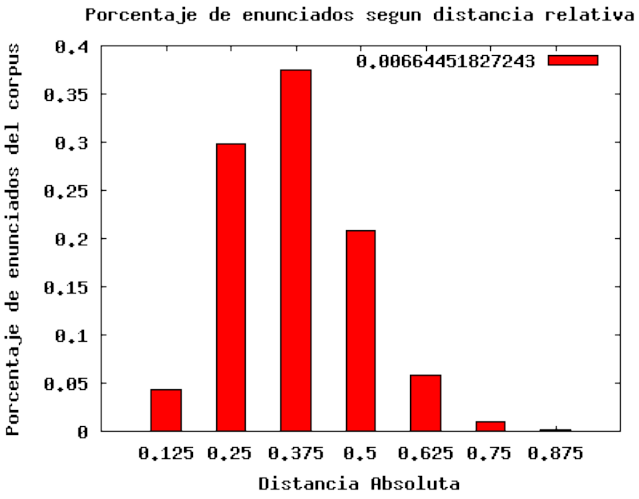


Figura 21: Histograma distancias relativas entre enunciados completos

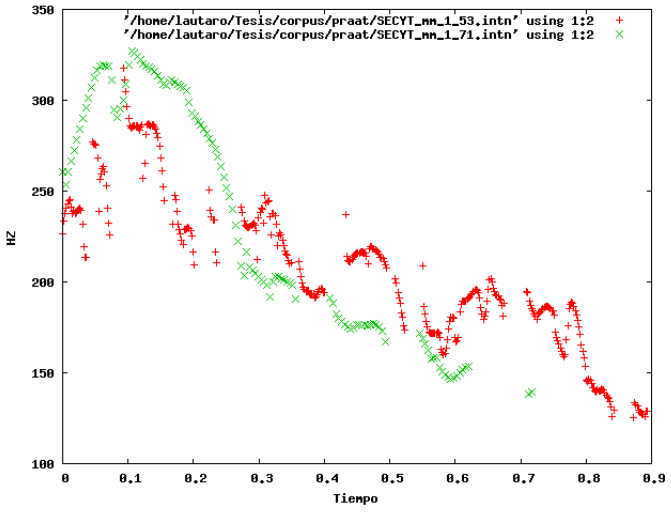


Figura 22: Contornos entonacionales con alto grado de similitud

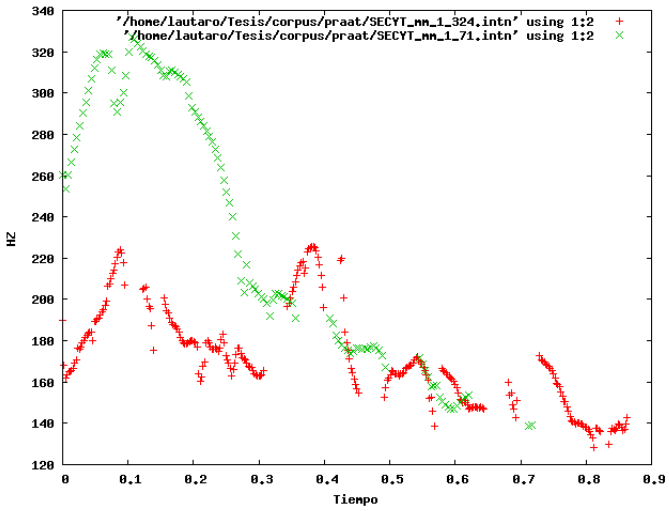


Figura 23: Contornos entonacionales con similitudes y diferencias

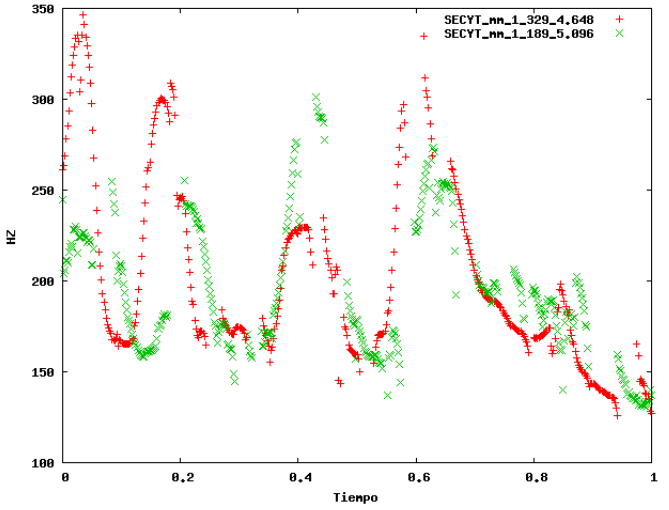


Figura 24: Contornos entonacionales similares al comparar unidades (frases sintácticas)

A.2. Modelo sintáctico

A.2.1. Alineación de las anotaciones

Descripción del problema En el actual corpus existen etiquetas o anotaciones de diferentes tipos pertenecientes a ciertos enunciados que se no corresponden temporalmente con las anotaciones temporales en la transcripción léxica del enunciado. Por ejemplo se puede ver que en el siguiente enunciado que corresponde al archivo SECYT_mm_1_3 existen diferentes problemáticas

```
Tiempo Color Label
0.360793  121 Ayer
0.816239  121 el
0.966667  121 general
1.560568  121 /p
1.598558  121 cumplió
2.299982  121 ochenta
3.035447  121 años
3.591771  121 /p

Tiempo Color Label
0.362344  121 V
0.816484  121 E
0.966667  121 S
1.649469  121 O
2.299982  121 N
3.040095  121 S
3.592451  121 /p
```

y las anotaciones de tipo POS.

Durante el experimento se buscó empíricamente un α para el cual dado un tiempo t al cual le corresponde una anotación léxica, exista un tiempo t' tal que $|t' - t| < \alpha$ y la etiqueta asociada se corresponda con la anotación léxica. En la figura 25 se puede ver en promedio el porcentaje de pérdida de información por archivo en función de α

A.3. Clustering

Durante los experimentos de clustering se utilizaron diferentes cotas a la hora de determinar la cantidad máxima de conjuntos a generar por métodos utilizados. Sin embargo se creyó prudente trabajar con un máximo de 10 conjuntos a la hora de realizar clustering.

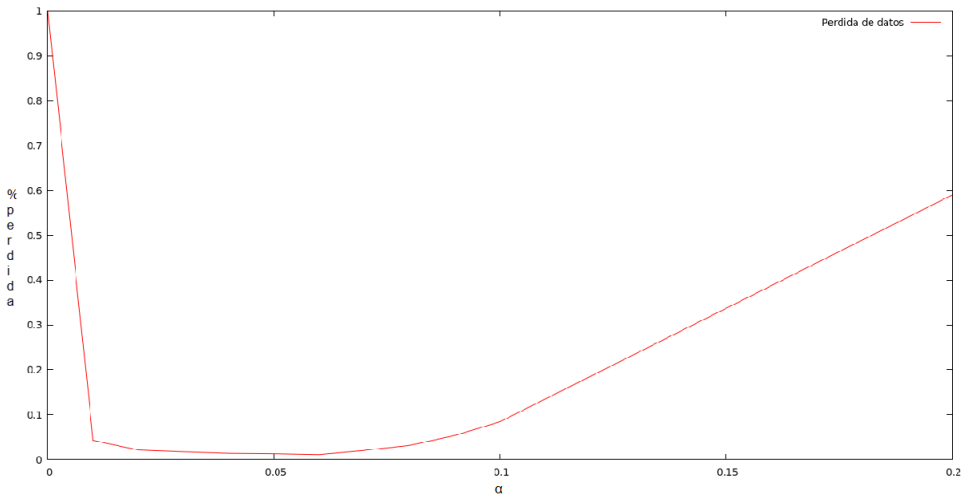


Figura 25: Pérdida de datos por alineación

Referencias

[BH94] Mary E. Beckman and Julia Hirschberg. The ToBI annotation conventions. *Ohio State Univ.*, 1994.

[Bri92] E. Brill. A simple rule-based part of speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 152–155. Association for Computational Linguistics, 1992.

[CDM+91] A. Colorni, M. Dorigo, V. Maniezzo, et al. Distributed optimization by ant colonies. In *Proceedings of the first European conference on artificial life*, volume 142, pages 134–142, 1991.

[FH84] H. Fujisaki and K. Hirose. Analysis of voice fundamental frequency contours for declarative sentences of japanese. *Journal of Acoustic Society*, 5(4):233–242, 1984.

[FR95] T.A. Feo and M.G.C. Resende. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, 6(2):109–133, 1995.

[Fra57] A.S. Fraser. Simulation of Genetic Systems by Automatic Digital Computers VI. Epistasis. *Australian Journal of Biological Sciences*, 13(2):150–162, 1957.

[Fuj94] Sumio Ohono Kei ichi Nakamura Miguelina Guirao Jorge Gurlekian Fujisaki, Hiroya. Analysis of accent and intonation in spanish based on a quantitative model. In *Proc. of 3rd International Conference on Spoken Language Processing (ICSLP 94)*, pages 355–358, 1994.

[Fuj02] Hiroya Fujisaki. Modeling in the study of tonal features of speech with application to multilingual speech synthesis. In *Joint International Conference of SNLP and Oriental COCOSDA*, pages 1–10, 2002.

[GRCT01] J. Gurlekian, H. Rodríguez, L. Colantoni, and H. Torres. Development of a prosodic database for an argentine spanish text to speech system. In *IRCS Workshop on Linguistic Databases*, Philadelphia, PA, 2001.

[GS62] D. Gale and L.S. Shapley. College admissions and the stability of marriage. *The American Mathematical Monthly*, 69(1):9–15, 1962.

[Hir93] Julia Hirschberg. Pitch accent in context: predicting intonational prominence from text. *Artificial Intelligence*, 63:305–340, 1993.

[HTF01] T. Hastie, R. Tibshirani, and J.H. Friedman. *The elements of statistical learning: data mining, inference, and prediction.* Springer Verlag, 2001.

[Lev66] V.I. Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet Physics Doklady*, volume 10, pages 707–710, 1966.

[Mix95] Hansjoerg y Hiroya Fujisaki Mixdorff. A scheme for a model-based synthesis by rule of f0 contours of german utterances. In *4th European Conference on Speech Communication and Technology (EUROSPEECH 95)*, pages 1823–1826, 1995.

[Mix00] H. Mixdorff. A novel approach to the fully automatic extraction of fujisaki model parameters. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 00)*, pages 1281–1284, 2000.

[MRSC08] C.D. Manning, P. Raghavan, H. Schütze, and Ebooks Corporation. *Introduction to information retrieval*, volume 1. Cambridge University Press Cambridge, UK, 2008.

[Nak98] Christine H. Nakatani. Constituent-based accent prediction. In *Proc of ACL*, 1998.

[PBH94] John F. Pitrelli, Mary E. Beckman, and Julia Hirschberg. Evaluation of prosodic transcription labeling reliability in the ToBI framework. In *Proc. of the International Conference of Spoken Language Processing (ICSLP)*, pages 123–126, 1994.

[PH00] S. Pan and J. Hirschberg. Modeling local context for pitch accent prediction. In *Proc of the 38th Annual Meeting on Association for Computational Linguistics*, pages 233–240. Association for Computational Linguistics Morristown, NJ, USA, 2000.

[R+96] A. Ratnaparkhi et al. A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1, pages 133–142, 1996.

[RO96] K. Ross and M. Ostendorf. Prediction of abstract prosodic labels for speech synthesis. *Computer Speech & Language*, 10(3):155–185, 1996.

[Sal71] G. Salton. The smart retrieval systemexperiments in automatic document processing. 1971.

[Sha51] C.E. Shannon. Prediction and entropy of printed English. *Bell System Technical Journal*, 30(1):50–64, 1951.

[SN+87] N. Saitou, M. Nei, et al. The neighbor-joining method: a new method for reconstructing phylogenetic trees. *Mol Biol Evol*, 4(4):406–425, 1987.

[Spr94] Richard Sproat. English noun-phrase accent prediction for text-to-speech. *Computer Speech and Language*, 8(2):79–94, 1994.

[Sun02] Xuejing Sun. Pitch accent prediction using ensemble machine learning. In *ICLSP*, 2002.

[SWY75] G. Salton, A. Wong, and C.S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, 1975.

[TG04] H.M. Torres and J.A. Gurlekian. Automatic determination of phrase breaks for argentine spanish. In *International Conference on Speech Prosody*, Nara, Japan, 2004.

[Tor08] Humberto M. Torres. *Generación Automática de la Prosodia para un Sistema de Conversión de Texto a Habla*. PhD thesis, Facultad de Ingeniería, Universidad de Buenos Aires, 2008.

[War63] J.H. Ward. Hierarchical grouping to optimize an objective function. *Journal of the American statistical association*, 58(301):236–244, 1963.

[WF99] I.H. Witten and E. Frank. *Weka: Practical machine learning tools and techniques with Java implementations*. University of Waikato. Dept. of Computer Science, 1999.