

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN
KHOA KHOA HỌC MÁY TÍNH**



UIT

**BÁO CÁO CUỐI KÌ
HỆ THỐNG TÌM KIẾM BÀI HÁT**

Giảng viên : Ts. Nguyễn Vinh Tiệp

**Sinh viên : Nguyễn Hữu Doanh - 18520606
Nguyễn Minh Châu - 18520519
Tô Viết Anh - 18520471**

Lớp : CS336.L12.KHCL

Tháng 1/2021

Mục lục

LỜI MỞ ĐẦU.....	3
PHẦN 1: CƠ SỞ LÝ THUYẾT.....	4
1.1 Dữ liệu âm thanh.....	4
1.2 Phương pháp trích chọn đặc trưng truyền thống.....	5
PHẦN 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG	6
2.1 Mô tả dữ liệu	6
2.2 Keyword Retrieval.....	6
2.2.1 Giới thiệu	6
2.2.2 Pipeline	7
2.2.3 Kết quả thử nghiệm	7
2.3 Lyrics Retrieval.....	8
2.3.1 Giới thiệu	8
2.3.2 Vấn đề khi sử dụng độ đo TF-IDF khi tìm bài hát dựa trên đoạn lời.....	9
2.3.3 Pipeline	11
2.3.4 Linked Point	12
2.3.5 Kết quả thử nghiệm	14
2.3.6 Đánh giá.....	15
2.4 Melody Retrieval.....	15
2.4.1 Giới thiệu	15
2.4.2 Pipeline	16
2.4.3 Feature Extraction - MFCC.....	16
2.4.4 Khảo sát	17
2.4.5 Small World Graph	18
2.4.6 Hierarchical Navigable Small World.....	22
PHẦN 3: GIAO DIỆN HỆ THỐNG VÀ DEMO	26
3.1 Giao diện hệ thống.....	26
3.2 Demo	27
3.2.1 Chức năng Search lyric	27
3.2.2 Chức năng Search song	28
3.2.3 Chức năng Query audio	28
References.....	30

LỜI MỞ ĐẦU

Xin cảm ơn khoa Khoa học máy tính và thầy **Nguyễn Vinh Tiệp** đã cho nhóm có cơ hội được biết và tìm hiểu thêm về chủ đề Xây Dựng Hệ Thống Tìm Kiếm Bài Hát. Từ những kiến thức nền tảng được truyền tải ở mỗi buổi học, nhóm đã tìm tòi, học hỏi thêm các kiến thức mới và sau đó đã hoàn thành báo cáo về **Hệ thống Tìm Kiếm Bài Hát**. Trong quá trình thực hiện, những sai sót là không thể tránh khỏi. Chính vì vậy, nhóm mong nhận được những ý kiến góp ý từ giảng viên để báo cáo được hoàn thiện hơn.

Âm nhạc là một nhu cầu giải trí đóng vai trò quan trọng của hầu hết tất cả mọi người. Từ xa xưa thế giới đã có các thể loại nhạc xưa như thể loại nhạc đồng quê, nhạc miền tây... nói chung và đất nước ta nói riêng như ca trù, dân ca, quan họ, ... Hiện nay, nhiều thể loại nhạc mới cũng đã ra đời để có thể phục vụ cho nhu cầu giải trí của người dân, nổi tiếng như các thể loại nhạc Pop, Rock, R&B, Rap, ... Điều này đi song hành với công nghệ 4.0 khiến cho việc các ứng dụng nhạc trên xuất hiện ngày càng nhiều. Để có thể phục vụ nhu cầu của mọi người thì việc tìm kiếm nhạc cũng đóng một vai trò rất lớn trong các ứng dụng nhạc. Vì thế việc thiết kế một hệ thống tìm kiếm nhạc phải đáp ứng nhiều yếu tố như về tốc độ tìm kiếm, độ chính xác.

Giả sử tình huống, khi bạn đang ngồi uống cafe, đột nhiên chủ quán bật lên bài hát rất hay và bạn cũng rất thích. Nhưng giờ hỏi chủ quán thì ngại quá nên bạn chỉ có thể cố gắng ghi âm một đoạn ngắn trong bài hát đấy để về nhà tìm sau. Nhưng chẳng may nếu đó là một bài hát không lời thì lúc đấy phải làm sao. Đó là lý do trong bài báo cáo này nhóm sẽ giới thiệu về một hệ thống tìm kiếm bài hát mà nhóm đã phát triển sử dụng những phương pháp khác nhau, đồng thời cải thiện và tối ưu hóa độ chính xác và thời gian tìm kiếm nhằm đáp ứng được nhu cầu của người sử dụng hiện nay.

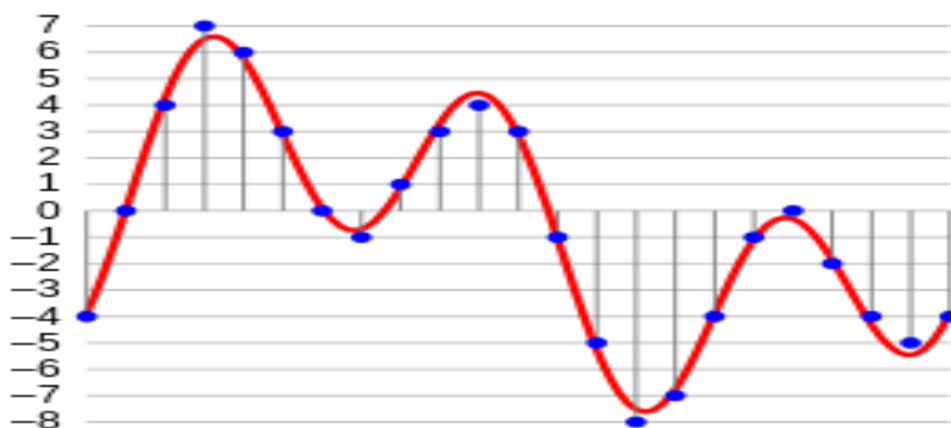
PHẦN 1: CƠ SỞ LÝ THUYẾT

1.1 Dữ liệu âm thanh

Một cách trực tiếp hay gián tiếp nào đó chúng ta vẫn tương tác với các dạng âm thanh hàng ngày. Đơn giản như tiếng lách cạch khi người bạn bên cạnh gõ bàn phím cũng làm bạn khó chịu hay tiếng của crush vừa mới xuất hiện trong đầu đã làm bạn sững đến mức phát điên cả ngày hôm đó. Tất cả những thứ đó được coi là dữ liệu dạng âm thanh. Tuy nhiên ở góc độ của máy tính chúng ta có thể hiểu dữ liệu dạng âm thanh là một chuỗi các biên độ kèm thời gian tương ứng tức là dữ liệu dạng âm thanh có dạng sóng.

Trên thực tế dữ liệu mà chúng ta thu được từ âm thanh là một dữ liệu phi cấu trúc, có nghĩa là chúng ta không thể biết được cấu trúc rõ ràng của nó. Chính vì thế nếu bạn bị người yêu mắng mà bạn cũng muốn chiếc máy tính của bạn hiểu được thì cần phải chuyển đổi tiếng đó thành dạng biên độ tương ứng với âm thanh như trên. Tuy nhiên một câu chuyện xảy ra là chúng ta không thể nào lưu hết được tần số theo trục thời gian (vì nó là vô hạn). Thay vào đó người ta nghĩ ra việc xấp xỉ hoá chúng, tức là khoảng 0.005s thì lưu lại giọng của cô người yêu bạn một lần. Do đặc tính của tai người (nghe người yêu mắng nhiều thành quen) bạn cũng sẽ không nhận ra được các âm thanh đó là rời rạc. Người ta gọi đó là kỹ thuật lấy mẫu âm thanh - sampling of audio data và việc bạn lấy bao nhiêu mẫu trong một giây - có thể là 40000 hoặc 16000 chẳng hạn được gọi là sampling rate.

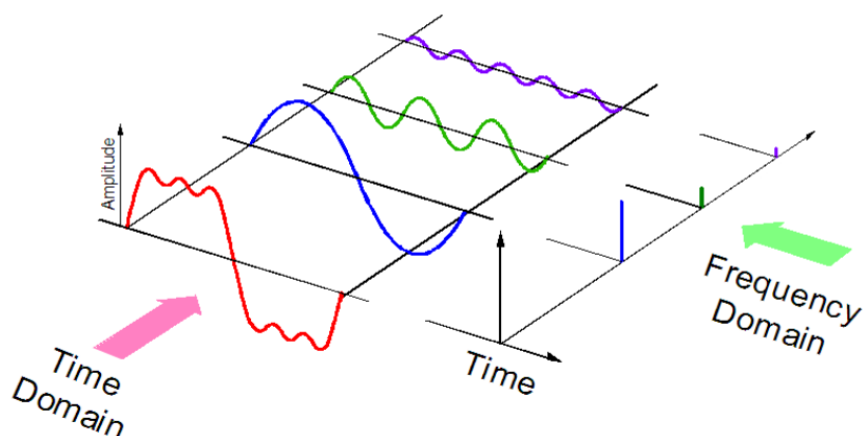
Chúng ta có thể hình dung nó như sau, trong đó các biên độ (amplitude) được thể hiện dưới hình sóng uốn lượn.



Tương ứng với mỗi điểm thời gian t_i chúng ta sẽ có một biên độ a_i và cứ như thế, âm thanh được tạo ra nhờ sự thay đổi các biên độ theo thời gian.

Có một cách khác để biểu diễn dữ liệu dạng âm thanh này đó chính là biểu diễn nó theo miền tần số. Hãy tưởng tượng rằng chúng ta chia giọng nói đánh thép của cô người yêu thành các phổ tần số khác nhau tương ứng theo thời gian, đồng nghĩa với việc chúng ta cần nhiều điểm để biểu diễn dữ liệu tại một thời điểm nhất định. Tất nhiên là tốc độ lấy mẫu **sampling rate** phải càng cao càng tốt. Chúng ta có thể tham khảo hình vẽ sau:

The central idea



Ở đây, chúng ta sẽ tách một tín hiệu âm thanh thành 3 tín hiệu thuần khác nhau, giờ đây có thể được biểu diễn dưới dạng 3 giá trị duy nhất trong miền tần số.

1.2 Phương pháp trích chọn đặc trưng truyền thống

Trong cách tiếp cận truyền thống, các vector đặc trưng của tín hiệu âm thanh được xây dựng từ các đặc trưng vật lý của âm thanh như độ to, độ cao, năng lượng, phổ tần số, ... Gần đây, một số nghiên cứu trên thế giới tập trung vào một cách tiếp cận khác, trong đó áp dụng các kiến thức về xử lý tín hiệu âm thanh, về phân tích mô hình tạo âm thanh, mô hình cảm thụ âm thanh của con người có thể giúp việc tính toán vector đặc trưng âm thanh được chính xác và hạn chế tối đa thông tin dư thừa.

PHẦN 2: PHÂN TÍCH VÀ THIẾT KẾ HỆ THỐNG

Trong đồ án nhóm sẽ cài đặt ba module để tìm kiếm bài hát:

1. Keyword Retrieval: Tìm kiếm bài hát thông qua từ khóa.
2. Lyrics Retrieval: Tìm kiếm bài hát thông qua đoạn lời.
3. Melody Retrieval: Tìm kiếm bài hát thông qua giai điệu.

2.1 Mô tả dữ liệu

Dataset bao gồm:

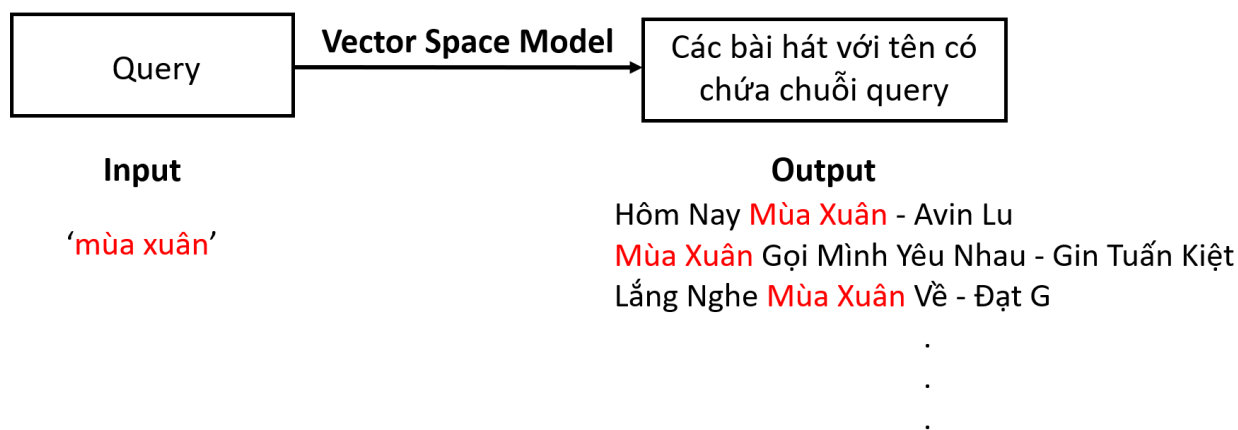
- 2000 file âm thanh định dạng .mp4.
- 2000 file lời nhạc định dạng .txt.
- 2000 file ảnh album định dạng .jpg.

2.2 Keyword Retrieval

2.2.1 Giới thiệu

Keyword Retrieval là hệ thống dùng để tìm kiếm bài hát qua từ khóa mà người dùng nhập vào. Đầu vào của hệ thống là một chuỗi query. Đầu ra là danh sách các bài hát với **tên bài hát** có chứa chuỗi query. Ở hệ thống này nhóm sử dụng phương pháp Vector Space Model với độ đo TF-IDF.

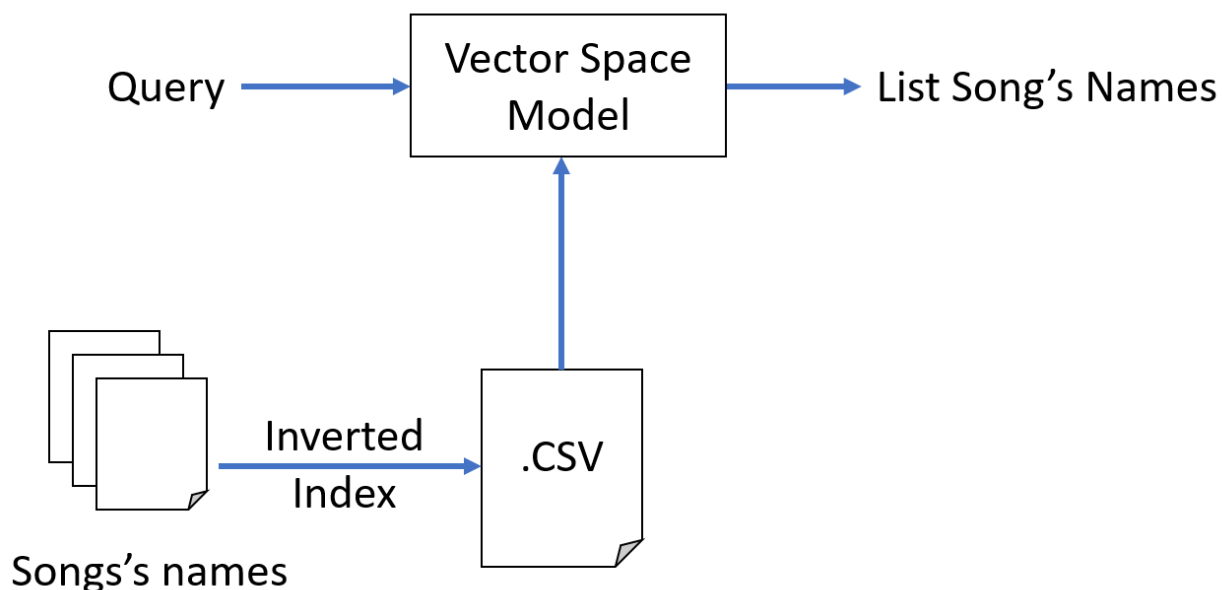
Hình dưới đây thể hiện tổng quan về hệ thống tìm kiếm bài hát dựa vào từ khóa:



Ở đây chuỗi query là chuỗi 'mùa xuân', khi đi qua mô hình Vector Space sẽ trả ra các bài hát với tên có chứa chuỗi 'mùa xuân'.

2.2.2 Pipeline

Pipeline này được dựa vào của phương pháp Vector Space Model có sử dụng Inverted Index.



Sử dụng Inverted Index (Chỉ mục ngược) giúp cho việc tìm kiếm trở nên nhanh hơn. Ở đây nhóm đã lưu chỉ mục ngược cho tên bài hát và lưu lại dưới dạng .csv và được lưu trữ dưới hệ thống server. Khi người dùng nhập vào chuỗi query là từ khóa sẽ được đưa qua mô hình Vector Space, mô hình sẽ sử dụng file .csv đã được lưu trước đó để tính TF-IDF cho từng từ có trong file .csv trong tất cả các bài hát và query, tính khoảng cách TF-IDF từng bài hát với query. Sau đó hệ thống sẽ lấy top k các bài hát có khoảng cách đến query ngắn nhất, trong đó $k \in N$ là số bài hát liên quan nhất hệ thống sẽ trả ra.

2.2.3 Kết quả thử nghiệm

Dưới đây là kết quả khi chuỗi query được đưa vào là chuỗi ‘người đàn ông’, kết quả sau khi trả ra là top 10 bài hát với tên bài hát liên quan với chuỗi ‘người đàn ông’ nhất.

Query = 'người đàn ông'

Result:

Người Đàn Ông - Phạm Trưởng

Người Đàn Ông Em Yêu Là Ai - Gia Minh

Đàn Ông Mộng Mị - Dương Ngọc Bích

Ông Trời Để Lạc Em Ở Đây - Trọng Hiếu

Người Yêu Em Và Người Em Yêu - Khánh Đơn

Anh Người Yêu - Cherie (Trâm Anh)

Người Oi Người Ở Đứng Về - Đức Phúc

Đã Từng Có Một Người Yêu Em - Bảo Kun

Về Nhà Thôi Người Anh Em - Nam Dương

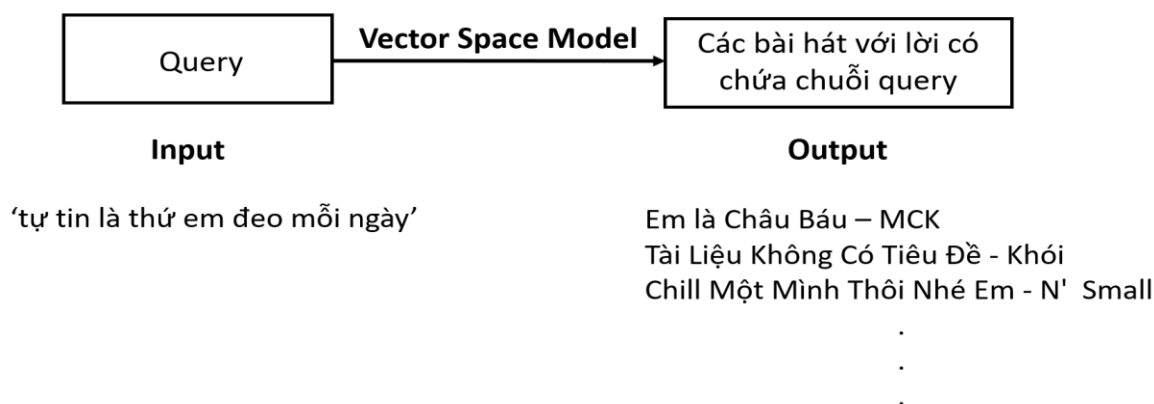
Ai Là Người Thương Em - Quân A

2.3 Lyrics Retrieval

2.3.1 Giới thiệu

Lyrics Retrieval là hệ thống dùng để tìm kiếm bài hát qua đoạn lời mà người dùng nhập vào. Đầu vào của hệ thống là một chuỗi query. Đầu ra là danh sách các bài hát với **lời bài hát** có chứa chuỗi query. Ở hệ thống này nhóm sử dụng phương pháp Vector Space Model với độ đo TF-IDF.

Hình dưới đây thể hiện tổng quan về hệ thống tìm kiếm bài hát dựa vào đoạn lời:



2.3.2 Vấn đề khi sử dụng độ đo TF-IDF khi tìm bài hát dựa trên đoạn lời

Sau khi sử dụng mô hình Vector Space sử dụng độ đo TF-IDF cho việc tìm kiếm các bài hát dựa trên đoạn lời, hệ thống cho ra các kết quả không như mong đợi, cụ thể qua ví dụ dưới đây, chuỗi đầu vào (query) là chuỗi ‘tự tin là thứ em đeo mỗi ngày’ được trích ra từ bài hát ‘Em là Châu Báu’ của MCK nhưng kết quả nhận được lại cho ra 10 bài hát khác với bài hát đã được trích.

‘Em là Châu Báu – MCK’ → query = 'tự tin là thứ em đeo mỗi ngày'

Result:

Chỉ Là Viễn Vong - Wansentai

Thôi Miên - CARA

Người Yêu Em Và Người Em Yêu - Khánh Đơn

Người Đứng Sau Hạnh Phúc - Don Hồ

Chẳng Gì Là Mãi Mãi - Châu Khải Phong

Cuộc Gọi - Phan Mạnh Quỳnh

Ngày Cô Đơn - Dương Phi Hoàng

Có Chắc Yêu Là Đây - Đoàn Gia Tuấn

Chill Một Mình Thôi Nhé Em - N' Small

Tự Nắm Tay Mình - Dương Hoàng Yến

Nhóm đã chọn bài hát đầu tiên mà hệ thống trả ra, cụ thể là bài ‘Chỉ là viễn vong’ của Wansentai và bài ‘Em là Châu Báu’ để phân tích nguyên nhân. Dưới đây là bảng so sánh số lượng từ thuộc query có trong 2 bài hát.

‘Em là Châu Báu – MCK’ → query = ‘tự tin là thứ em đeo mỗi ngày’	
Chỉ Là Viễn Vong - Wansentai	Em là Châu Báu – MCK
tự – 9 từ	tự – 1 từ
tin – 4 từ	tin – 1 từ
là – 32 từ	là – 12 từ
thứ – 1 từ	thứ – 1 từ
em – 47 từ	em – 34 từ
đeo – 0 từ	đeo – 1 từ
mỗi – 0 từ	mỗi – 1 từ
ngày – 12 từ	ngày – 3 từ
Tổng cộng : 105 từ	Tổng cộng : 54 từ

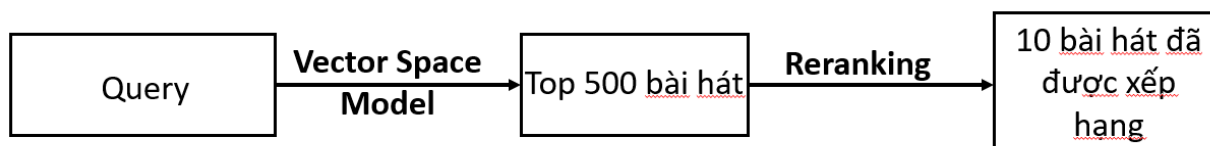
Theo bảng trên ta có thể thấy rõ số lượng các từ thuộc query có trong bài ‘Chỉ là viễn vong’ là 105 từ, gấp đôi số lượng từ thuộc query có trong ‘Em là Châu Báu’ chỉ với 54 từ.

Ta sẽ đi sâu hơn vấn đề bằng cách trích 2 đoạn lời của cả 2 bài hát trên, bảng dưới đây cho thấy với bài ‘Chỉ là viễn vong’ số lượng các từ thuộc query rất nhiều nhưng chúng rời rạc và không có tính liên kết so với bài ‘Em là Châu Báu’. Đây là lý do cho việc chỉ sử dụng mô hình Vector Space và độ đo TF-IDF không thể giải quyết được vấn đề tìm bài hát dựa trên đoạn lời.

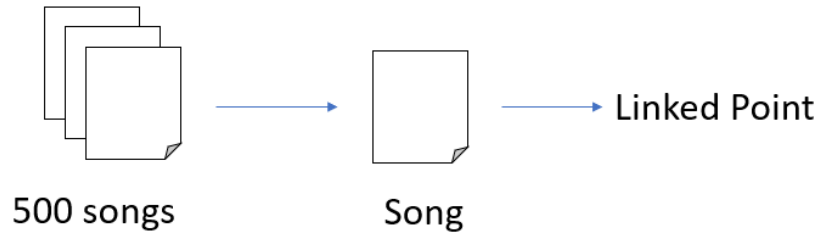
Chỉ Là Viễn Vong - Wansentai	Em là Châu Báu – MCK
<p>Là vì ta rồi mãi cũng xa Qua cơn say kia Vờ như anh vẫn luôn kề bên Đến đây thôi Vì biết ánh mắt kia chẳng phải là cho em Nên thôi xin đành Xin đành anh ơi bước đi Hơn là mãi ôm thấu đau Về những ngày ta đã Là vì em cứ ngỡ chiếc hôn ngày qua Là vì em cứ tin anh luôn nhìn về phía em Là vì em tự dối tự lừa chính em Và dù cho em có đi anh cũng chẳng ôm em?</p>	<p>Vì em lỡ quá yêu mình mất rồi (Oops) "Trần gian châu báu" không ai ngoài em Sha la la sha la la sha la la sha la la sha la la Em, tự tin là thứ em đeo mỗi ngày Và em "tỏa sáng-hơn-chữ-sáng" luôn này! Một-không-hai nếu không em thì ai? Sha la la sha la la sha la la sha la la sha la la Đi ngược lại với tất cả từ ngày xưa Em vẫn chất kể cả khi gầy hay quá cỡ Cá tính mạnh ăn hết mọi cá con và cá nhỏ Chuẩn mực phải đẹp đặt ra là để em phá vỡ Vẫn rất quý phái classy lady Ai bảo con gái không được crazy?</p>

2.3.3 Pipeline

Lý do chọn 500 bài hát giúp việc reranking trở nên nhanh hơn mà không cần phải xét hết tất cả các bài hát có trong database.



Từng bài hát trong 500 bài hát sẽ được gán một thuộc tính **Linked Point**, thuộc tính này được định nghĩa là độ liên kết giữa các từ thuộc query trong một bài hát.



2.3.4 Linked Point

Đoạn lyric được lấy ra trong 1 bài hát. Ban đầu ta khai báo biến **TLinked Point** và biến **Linked Point** có giá trị bằng 0. Dấu ngoặc nhọn **T** được xem là độ liên kết giữa từ đầu tiên với các từ tiếp theo. Ban đầu độ liên kết có giá trị bằng 2, tức từ đầu tiên kết nối với 1 từ tiếp theo trong đoạn lyric, cụ thể ở hình dưới đây từ đầu tiên ‘tự’ sẽ kết nối với từ ‘tin’ và được gán vào biến **a**. Vậy là biến **a** là biến chuỗi có giá trị là ‘tự tin’. Tiếp tục kết nối từ đầu tiên của đoạn lyric kết nối với từ liên kế và gán vào biến **b**. Biến **b** sẽ là biến chuỗi mang giá trị ‘tự tin’.

Query:	tự tin là thứ em đeo mỗi ngày
	<u> </u>
	a
Đoạn lyric:	tự tin là chính tôi
	<u> </u>
	b
	TLinked Point = 0
	Linked Point = 0

Ở hình dưới đây, biến **a** và biến **b** đều mang giá trị ‘tự tin’ nên **TLinked Point** sẽ được cộng lên 1.

Query:	tự tin là thứ em đeo mỗi ngày
	<u> </u>
	a
Đoạn lyric:	tự tin là chính tôi
	<u> </u>
	b
	TLinked Point = 1
	Linked Point = 0

Tiếp theo, ta tăng giá trị độ liên kết lên thành 3, tức từ đầu tiên sẽ liên kết với 2 từ tiếp theo thành 3 từ trong query và đoạn lyric và sau đó lần lượt gán cho biến **a** và biến **b**.

Query:	tự tin là thứ em đeo mỗi ngày
	a
Đoạn lyric:	tự tin là chính tôi
	b
TLinked Point = 1	
Linked Point = 0	

Ở đây biến **a** và biến **b** có giá trị ‘tự tin là’ nên biến **TLinked Point** được cộng 1 lên thành 2.

Query:	tự tin là thứ em đeo mỗi ngày
	a
Đoạn lyric:	tự tin là chính tôi
	b
TLinked Point = 2	
Linked Point = 0	

Độ liên kết được tăng lên thành 4. Biến **a** mang giá trị ‘tự tin là thứ’ và biến **b** mang giá trị ‘tự tin là chính’, hai biến không còn giống nhau nữa thì ta kết thúc quá trình kiểm tra giữa query và đoạn lyric này.

Query:	tự tin là thứ em đeo mỗi ngày
	a
Đoạn lyric:	tự tin là chính tôi
	b
TLinked Point = 2	
Linked Point = 0	

Sau khi kết thúc quá trình kiểm tra thì biến **Linked Point** sẽ được so sánh với biến **TLinked Point**, nếu giá trị của **Linked Point** nhỏ hơn giá trị của biến **TLinked Point** thì ta sẽ cập nhật giá trị cho **Linked Point** bằng với giá trị của **TLinked Point**, cụ thể ở ví dụ này **Linked Point** có giá trị là 0 được cập nhật lên thành 2.

Query:	tự tin là thứ em đeo mỗi ngày
	a
Đoạn lyric:	tự tin là chính tôi
	b
TLinked Point = 2	
Linked Point = 2	

2.3.5 Kết quả thử nghiệm

‘Em là Châu Báu – MCK’ → Query = ‘tự tin là thứ em đeo mỗi ngày’	
TF-IDF	TF-IDF + Linked Points
Result:	Result:
Chỉ Là Viễn Vong - Wansentai	Em Là Châu Báu - MCK.txt
Thôi Miên - CARA	Tài Liệu Không Có Tiêu Đề - Khói.txt
Người Yêu Em Và Người Em Yêu - Khánh Đơn	When You Go - KallawPie.txt
Người Đứng Sau Hạnh Phúc - Don Hồ	Chill Một Mình Thôi Nhé Em - N' Small.txt
Chẳng Gì Là Mãi Mãi - Châu Khải Phong	Take My Heart - Tommy.txt
Cuộc Gọi - Phan Mạnh Quỳnh	Tình Yêu Xa Xôi - JK Dương Tiên.txt
Ngày Cô Đơn - Dương Phi Hoàng	Thư Oi - Tăng Duy Tân.txt
Có Chắc Yêu Là Đây - Đoàn Gia Tuấn	Đang Bên Em Có Ai - Nozi.txt
Chill Một Mình Thôi Nhé Em - N' Small	Broken - OLDboy.txt
Tự Nắm Tay Mình - Dương Hoàng Yến	Short Skirt - Niz.txt

2.3.6 Đánh giá

Cách đánh giá:

Ban đầu cho biến **tp** có giá trị là 0. Lặp ngẫu nhiên 2000 bài hát, trong từng bài hát được chọn sẽ lấy ra ngẫu nhiên một đoạn lời làm input cho hệ thống, sau khi lấy output của hệ thống sẽ đem đi so sánh với tên bài hát được chọn, nếu đúng thì biến **tp** sẽ được cộng lên 1. Sau khi đã thử 2000 bài hát, ta lấy biến **tp** chia cho 2000 sẽ ra được accuracy của hệ thống.

Dưới đây là mã giả của cách đánh giá vừa nêu trên:

```
tp = 0
For i in range(2000):
    Chọn một lời bài hát ngẫu nhiên và đọc -> song.
    Lấy tên của song -> song_name.
    Lấy ngẫu nhiên 1 dòng trong song -> line.
    lyric(line) -> res # res là tên bài hát mà hệ thống tính được.
    if res == song_name:
        tp += 1
tp / 2000 -> acc
```

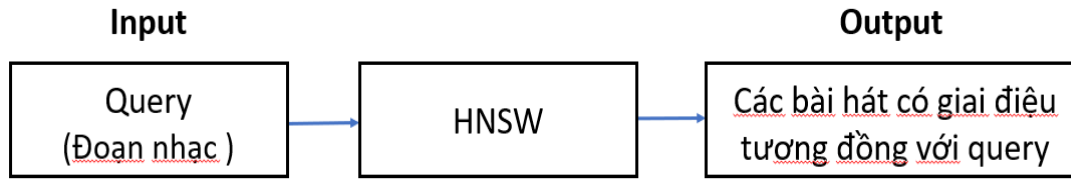
Sau khi chạy 2000 vòng lặp, nhóm nhận được kết quả “**Accuracy: 0.98**” cho thấy hệ thống đã thực hiện tốt việc truy vấn tìm kiếm lời bài hát.

2.4 Melody Retrieval

2.4.1 Giới thiệu

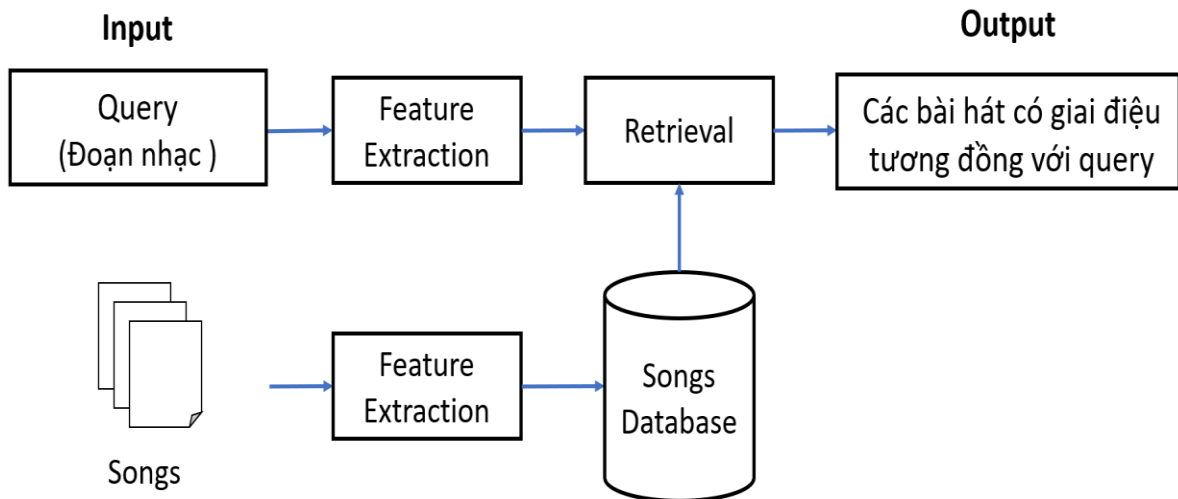
Melody Retrieval là hệ thống dùng để tìm kiếm bài hát qua giai điệu mà người dùng nhập vào. Đầu vào của hệ thống là một đoạn giai điệu. Đầu ra là danh sách các bài hát với **giai điệu** tương tự với đoạn query. Ở hệ thống này nhóm sử dụng phương pháp Hierarchical Navigable Small World (HNSW) để thực hiện quá trình tìm kiếm dựa vào các vector đặc trưng đã được trích xuất được từ file audio.

Hình dưới đây thể hiện tổng quan về hệ thống tìm kiếm bài hát dựa vào giai điệu:



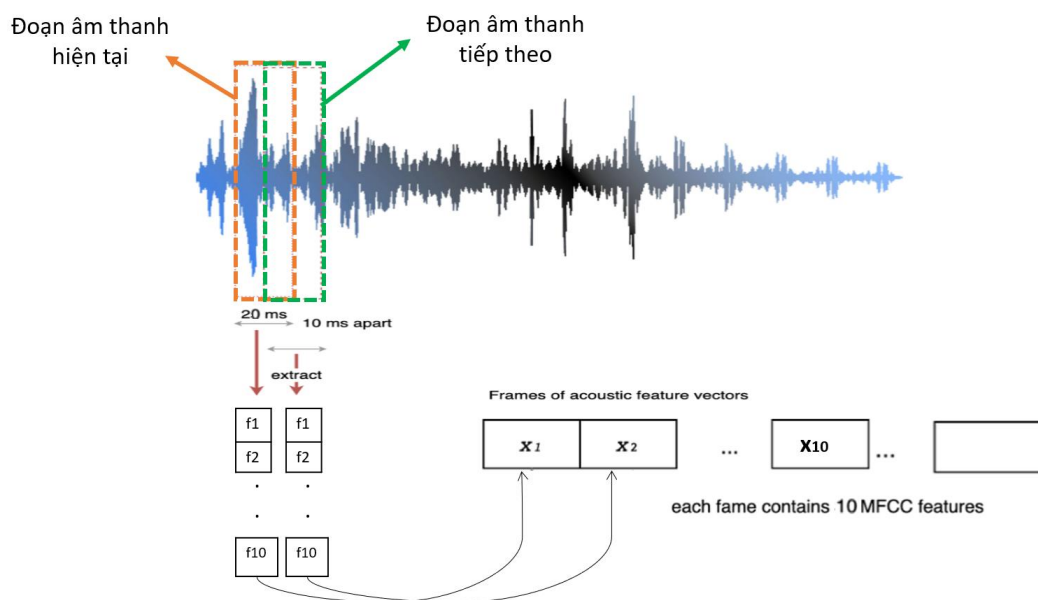
2.4.2 Pipeline

Pipeline của Melody Retrieval gần giống với pipeline của Keyword và Lyrics Retrieval. Đầu tiên ta trích xuất đặc trưng bài hát mà ta muốn truy vấn bằng phương pháp Mel Frequency Cepstrals Coefficients. Sau đó sẽ tìm kiếm các đặc trưng tương tự có trong database (chứa tất cả bài hát đã được trích xuất đặc trưng). Các hát nào có chứa nhiều đặc trưng tương tự nhất thì sẽ được xuất ra.



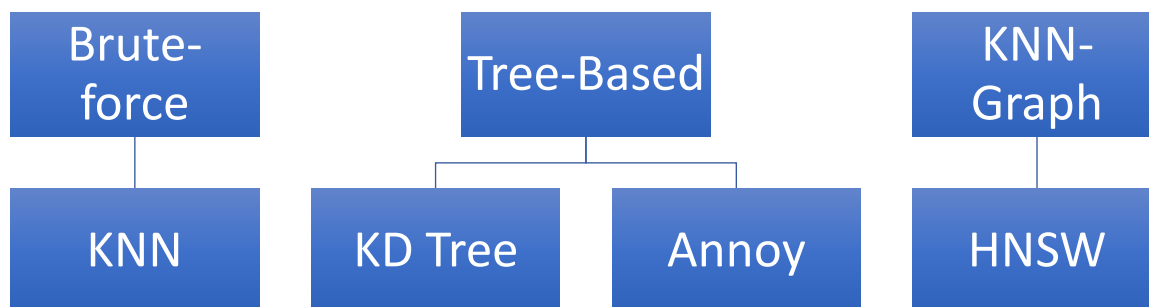
2.4.3 Feature Extraction - MFCC

Bài hát qua quá trình lấy mẫu ta sẽ thu được một chuỗi âm thanh dạng sóng, sau đó cắt chuỗi tín hiệu âm thanh đó thành các đoạn ngắn bằng nhau (20ms) với mỗi 20 ms âm thanh sẽ được trích xuất thành 10 đặc trưng và được gọi là 1 MFCC, 10 đặc trưng tiếp theo được trích xuất từ đoạn âm thanh 20 ms tiếp theo nhưng đoạn âm thanh này là đoạn âm thanh trước đó cộng thêm 10 ms tiếp theo. Việc lấy đoạn âm thanh tiếp theo overlap với đoạn âm thanh hiện tại giúp cho việc tìm kiếm đoạn giai điệu trở nên chính xác hơn. Như vậy thì 1 giây chúng ta sẽ sinh ra được 50 MFCC đặc trưng cho âm thanh. Tuy nhiên việc lấy 10 đặc trưng và thời gian lấy mẫu là 20ms dường như là quá nhỏ chưa đủ đặc trưng. Chính vì thế chúng ta sẽ gộp 10 MFCC này lại thành một vector 100 chiều. Đại diện cho mỗi đoạn âm thanh 200ms.



2.4.4 Khảo sát

Trong quá trình thực hiện đồ án này, nhóm đã có thực nghiệm chạy thử trên các phương pháp tìm kiếm phổ biến hiện nay để tìm kiếm nhạc theo giai điệu và dưới đây là cụ thể các phương pháp mà nhóm đã chạy:



Bảng khảo sát các phương pháp tìm kiếm trong Vector Space mà nhóm đã thực nghiệm

Trong quá trình thực nghiệm, nhóm đã rút ra được những điểm lợi và điểm yếu của các phương pháp tìm kiếm dựa vào tốc độ thực thi và độ chính xác.

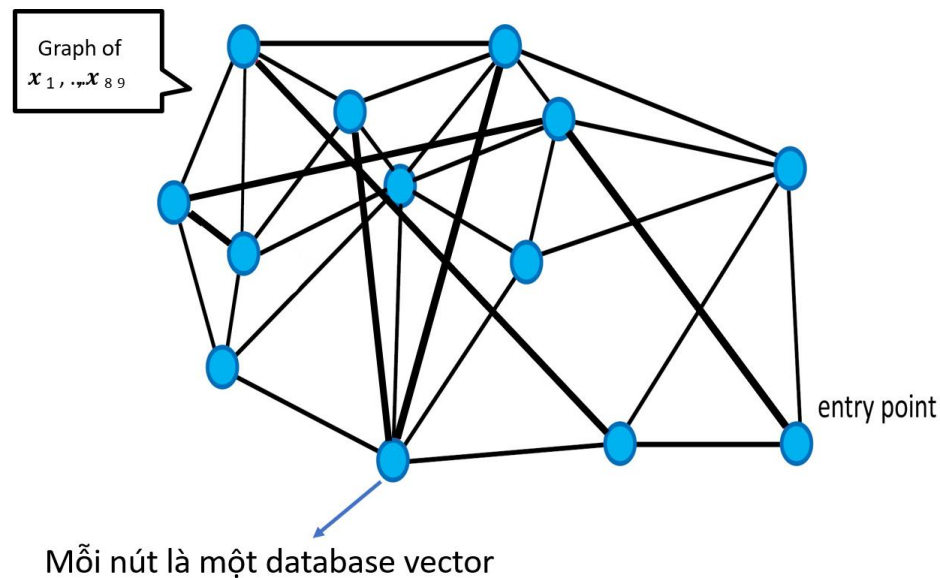
Phương pháp	Tốc độ	Độ chính xác
KNN	Rất chậm	Rất cao
KD Tree	Chậm	Cao
Annoy	Vừa	Vừa
HNSW	Nhanh	Cao

Bảng đánh giá các phương pháp tìm kiếm trong Vector Space mà nhóm đã thực nghiệm

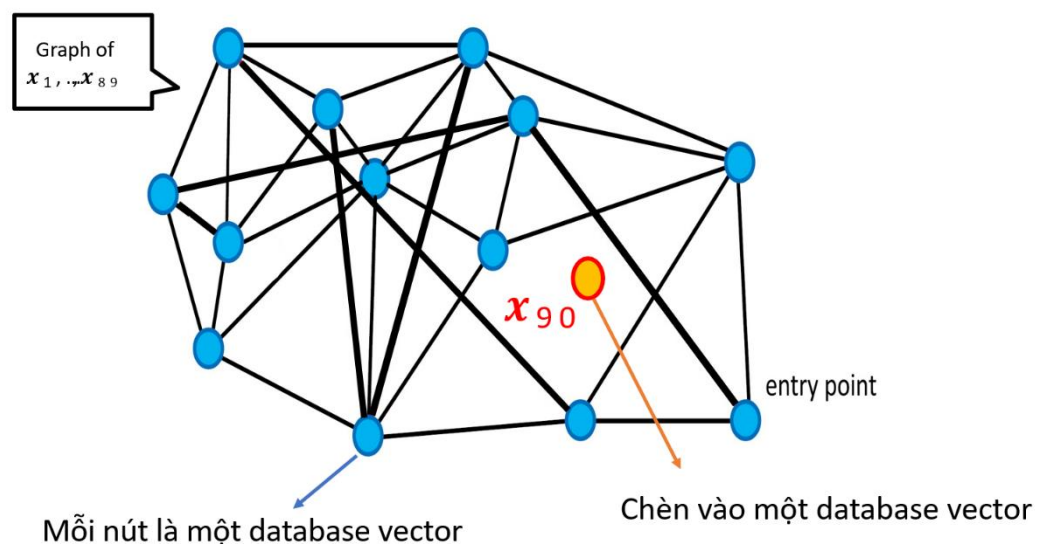
2.4.5 Small World Graph

2.4.5.1 Chèn thêm một nút mới vào đồ thị

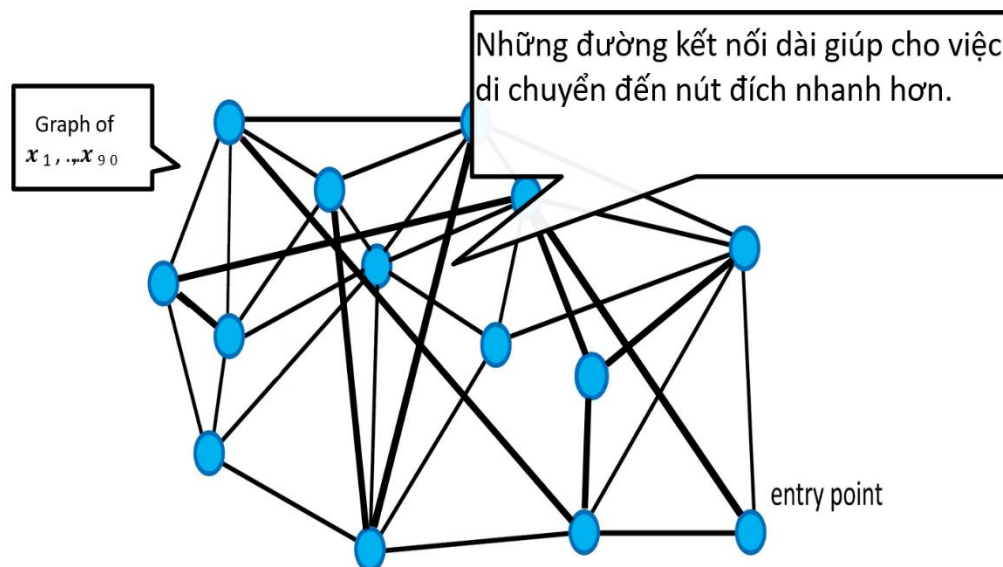
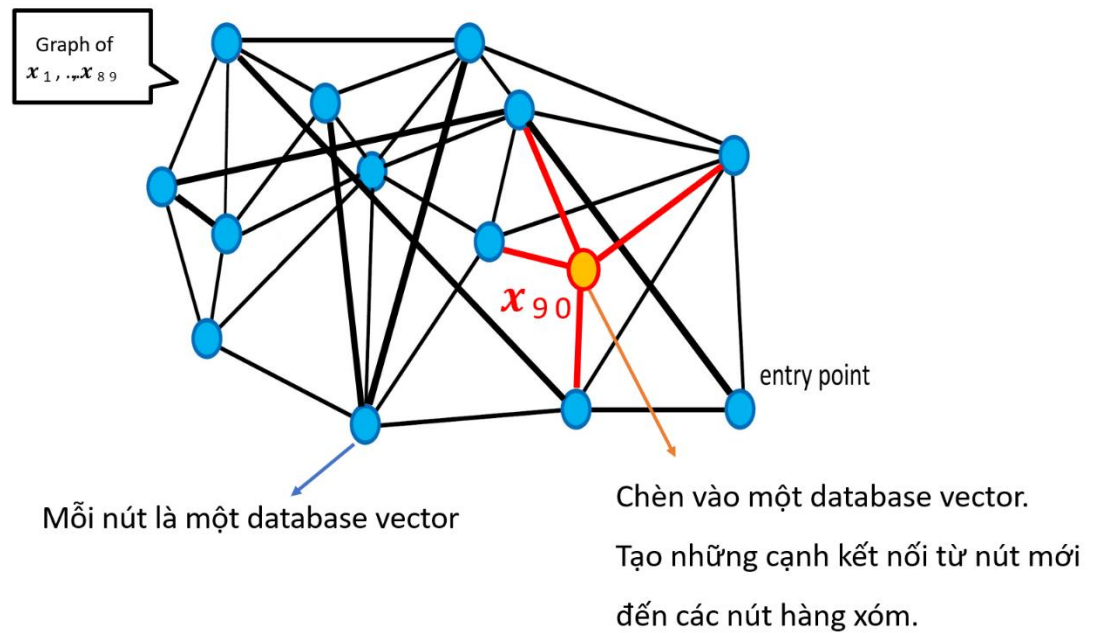
Các bài hát sau khi qua giai đoạn trích xuất đặc trưng sẽ được biểu diễn bằng các vector có chiều là 100. Sau đó các vector đó sẽ được xem như các nút màu xanh và được sắp xếp để tạo thành đồ thị như hình dưới hay còn được gọi là Small World Graph giúp cho việc tìm kiếm trở nên dễ dàng hơn do hệ thống có cấu trúc được thiết kế để việc tìm kiếm trở nên nhanh hơn.



Để thêm một nút vào ta sẽ biểu diễn vector đó trong không gian 100 chiều, tiếp theo ta sẽ tìm các nút lân cận (neighbors) của nút đó.

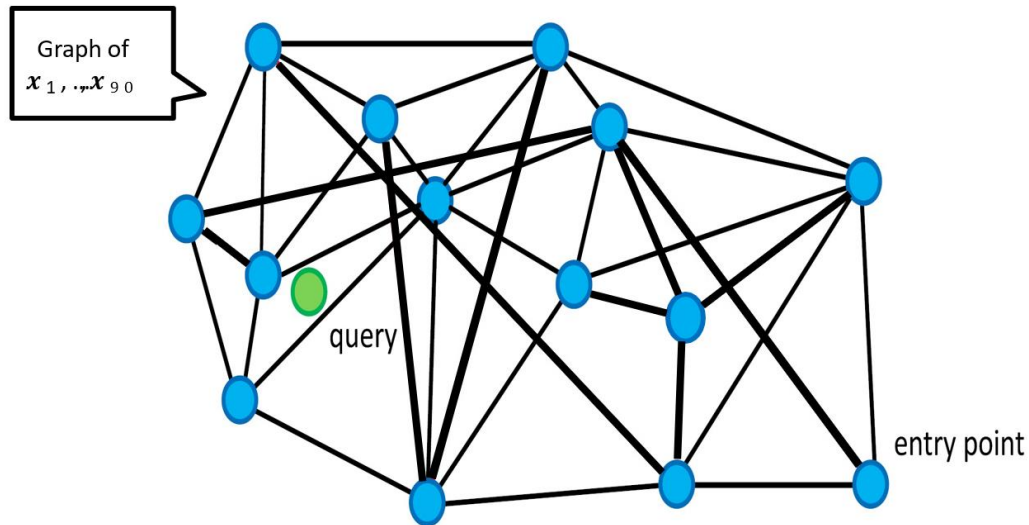


Bằng những độ đo tìm khoảng cách giữa hai nút, ta có thể tìm ra các nút lân cận, ở đây ta tìm được bốn nút lân cận với nút mới được thêm và đồ thị.

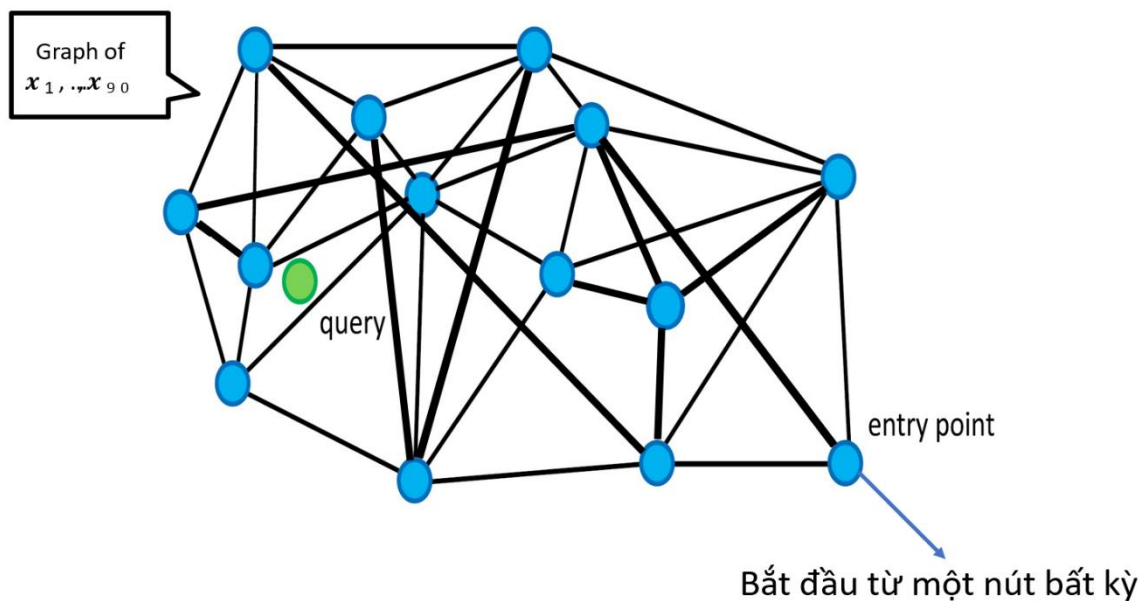


2.4.5.2 Tìm nút gần nhất với nút query

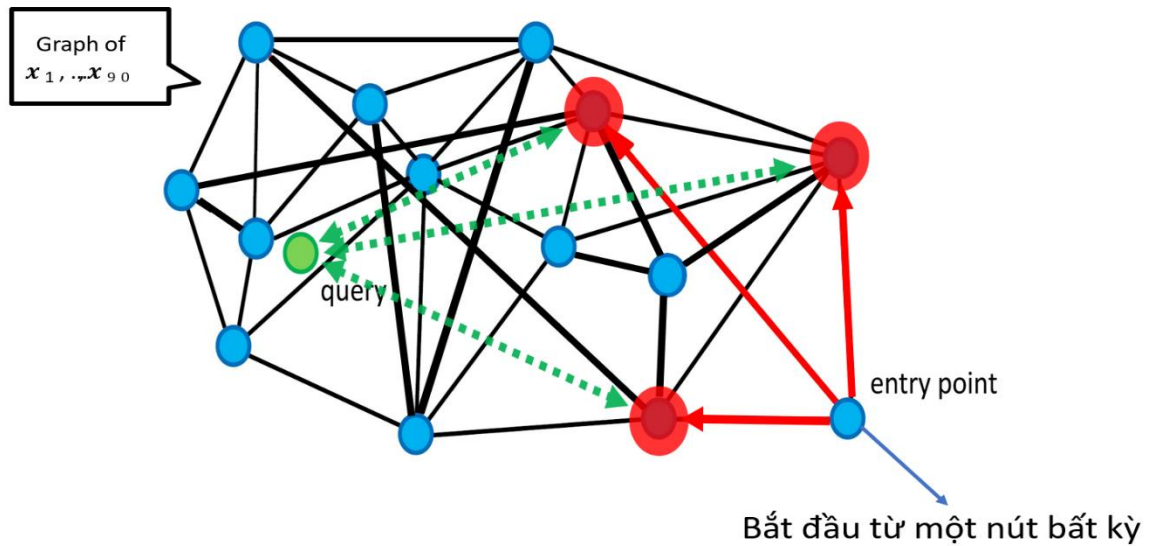
Ta có nút query được thêm vào đồ thị, nhiệm vụ của ta bây giờ là tìm nút gần với nút query nhất. Do đây là đồ thị 100 chiều và trường hợp đồ thị có nhiều nút thì việc tìm kiếm những điểm gần với nút query rất tốn thời gian và tài nguyên, nên việc tìm những nút lân cận của từng nút tạo điều kiện cho việc tìm kiếm dễ dàng và nhanh chóng hơn.



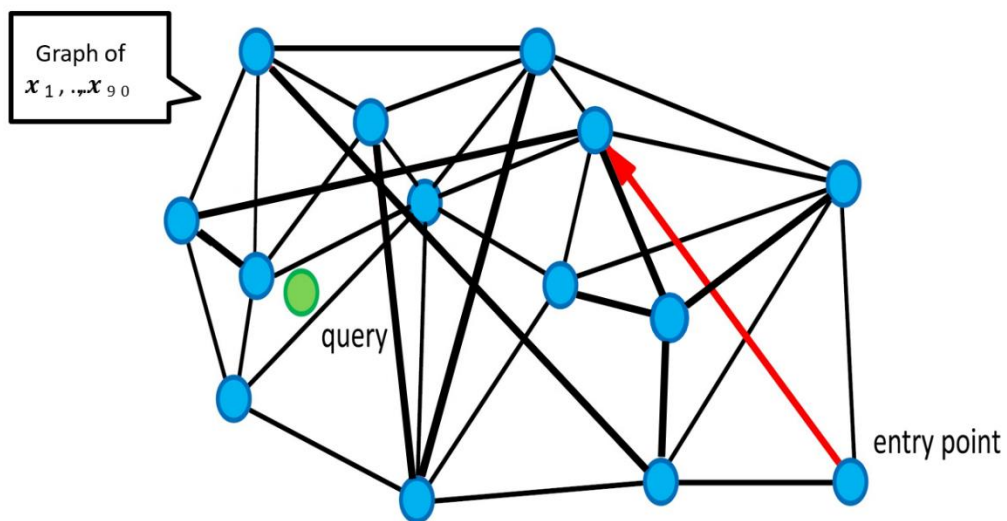
Ta chọn ngẫu nhiên một nút trong đồ thị và coi nó là Current Node.



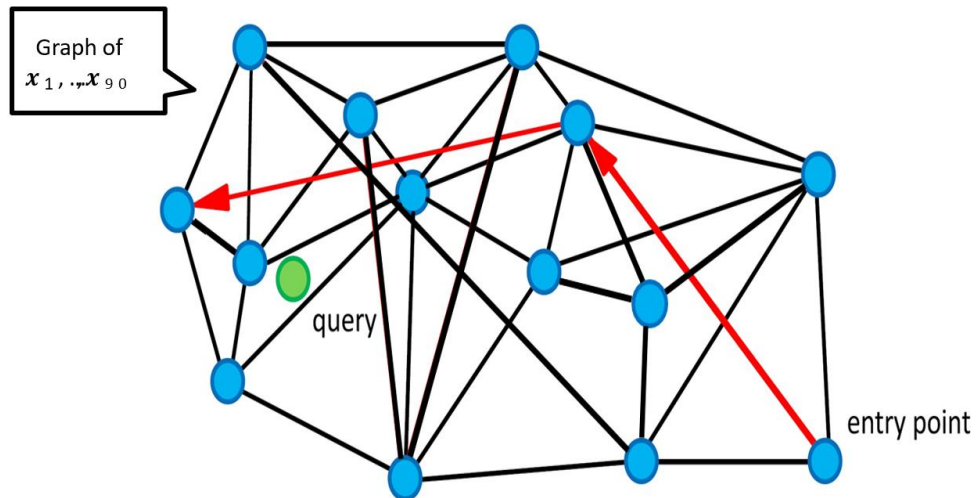
Từ nút đã được chọn ngẫu nhiên, ta có thể tìm được các nút lân cận do đã được thiết lập từ lúc xây dựng đồ thị, ở đây nút ta chọn được có ba nút lân cận được tô đỏ.



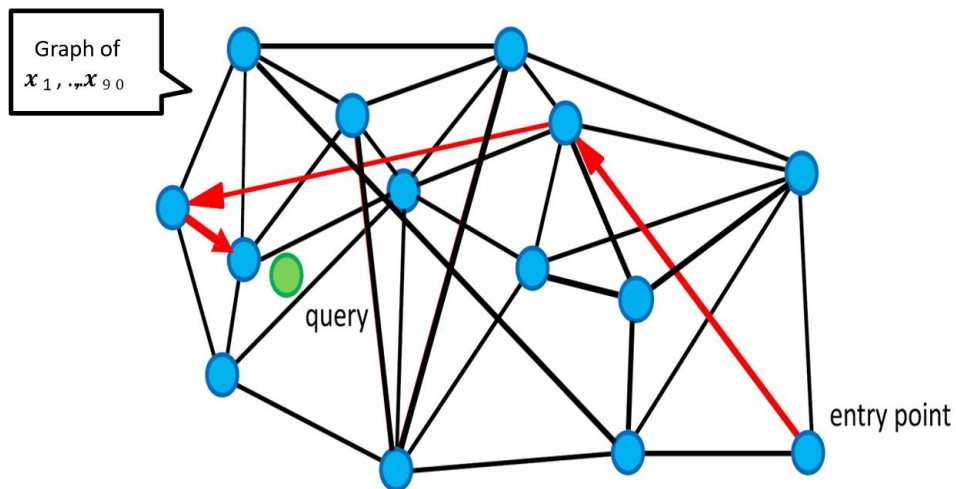
Có các nút lân cận ta tính khoảng cách giữa từng nút với nút query. Sau đó chọn nút gần với query nhất làm Current Node.



Tiếp tục quá trình tính khoảng cách giữa query với nút lân cận của Current Node.



Quy trình sẽ kết thúc khi Current Node là nút gần với nút query nhất trong đồ thị.



2.4.5.3 Ưu Nhược điểm

- Ưu điểm:
Việc tìm kiếm nút trong Small World Graph nhanh hơn nhiều so với kỹ thuật tìm kiếm Brute Force.
- Nhược điểm:
Nếu số lượng nút có trong Small World Graph đạt tới một con số khổng lồ đồng nghĩa với việc có một số nút có nhiều nút lân cận ảnh hưởng tới việc tính khoảng cách từ các nút lân cận với nút query.

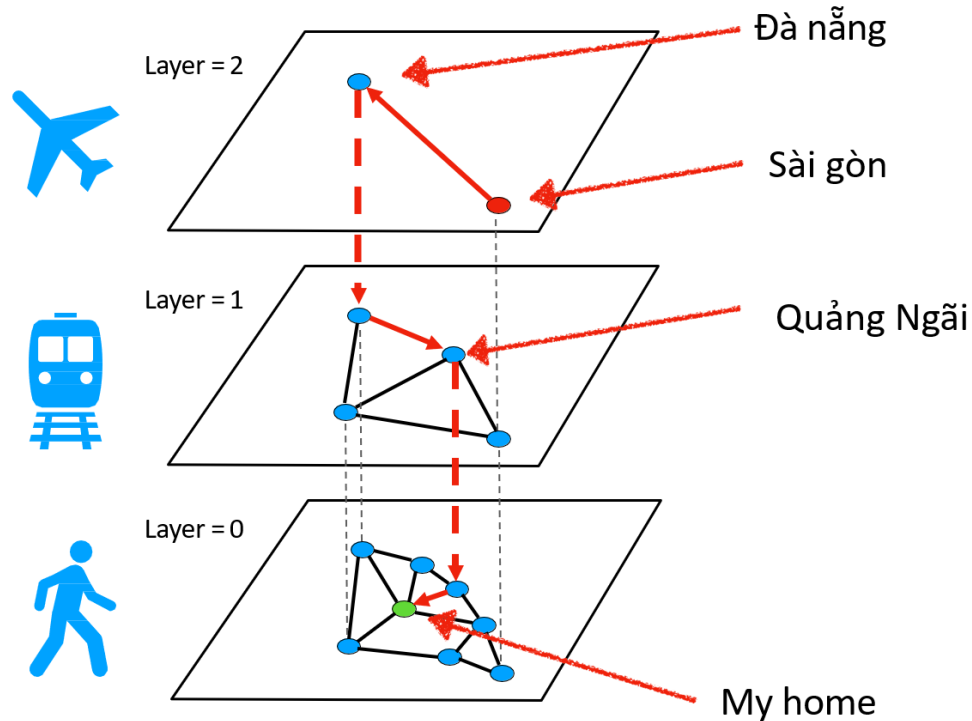
2.4.6 Hierarchical Navigable Small World

2.4.6.1 Giới thiệu

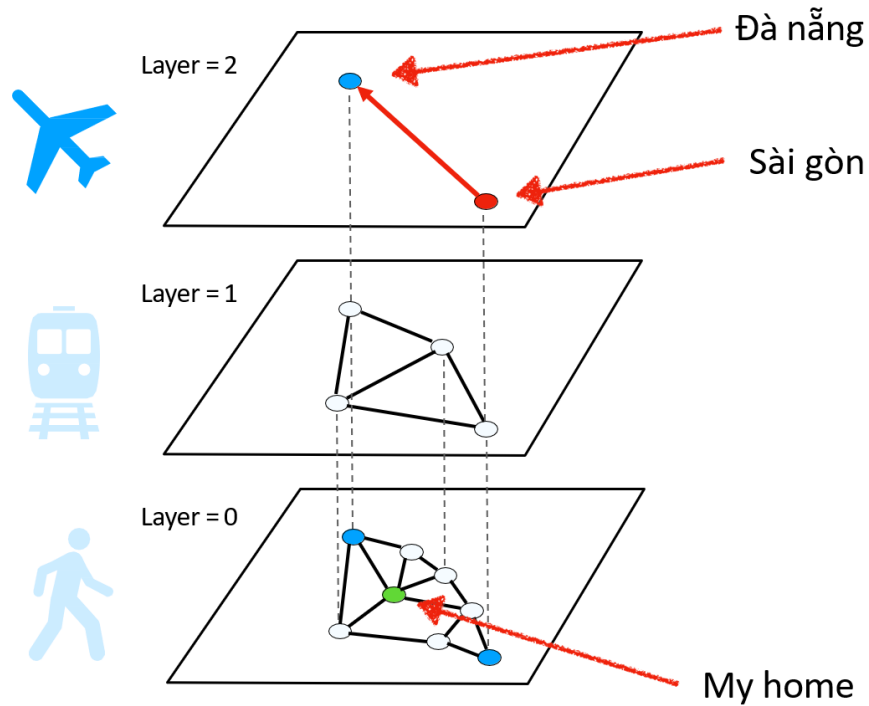
Hierarchical Navigable Small World (HNSW) là một đồ thị được kế thừa từ Small World Graph, nhưng thay vì các nút đều nằm trong một đồ thị thì HNSW sẽ chia ra làm nhiều tầng, mỗi tầng chứa một Small World Graph và mỗi tầng chứa các nút

tương ứng. Ở đây, các nút có nút lân cận có khoảng cách xa sẽ được bố trí ở tầng cao nhất, càng xuống tầng dưới thì khoảng cách giữa các nút và nút lân cận càng giảm.

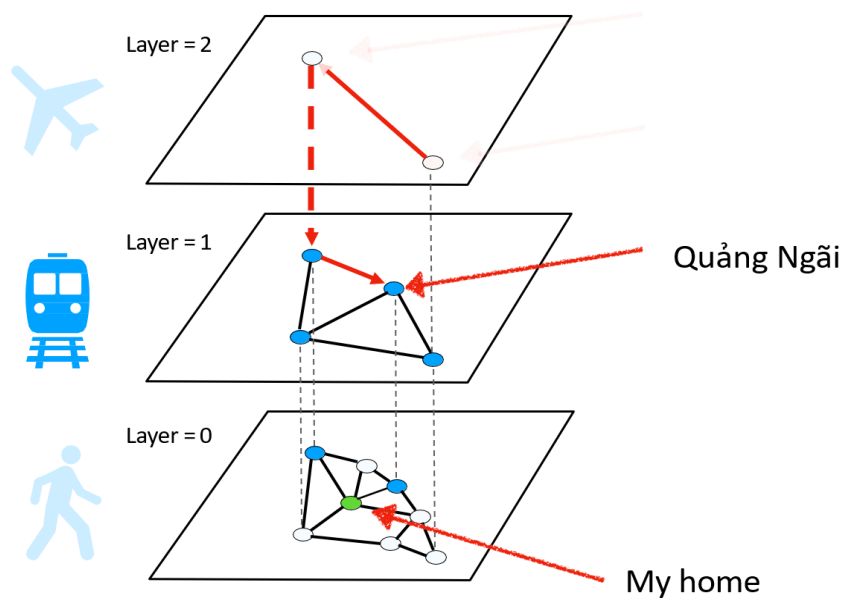
Lý do cho việc bố trí này dựa theo ưu điểm của Small World Graph, ví dụ Doanh là một sinh viên của trường UIT và Doanh đang ở **Sài Gòn**, tết gần đến và Doanh dự tính sẽ về nhà (**My home**) của mình ở **Quảng Ngãi**.



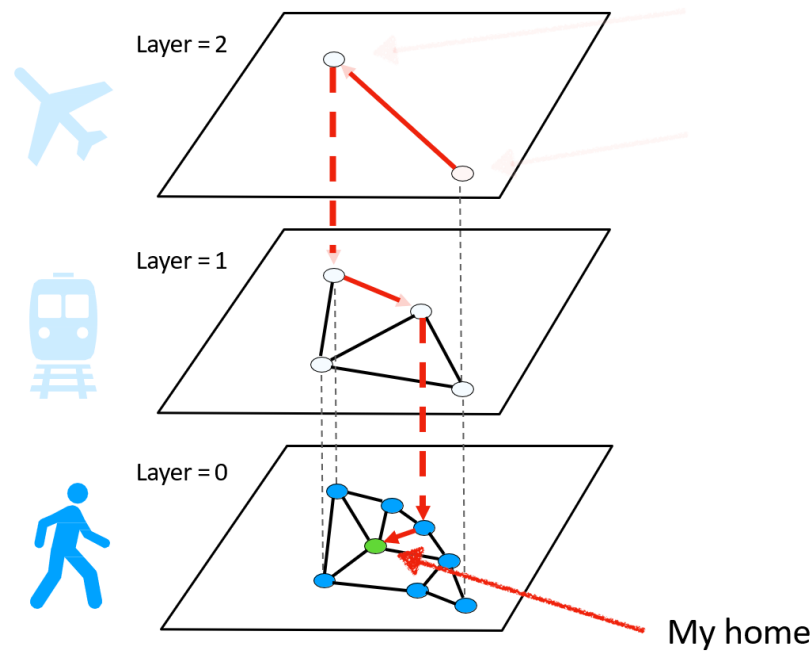
Để tối ưu hóa thời gian về nhà Doanh cần phải bắt máy bay đến **Quảng Ngãi**. Do ở **Quảng Ngãi** không có sân bay nên Doanh phải bắt máy bay đến tỉnh gần với **Quảng Ngãi** nhất và đó là **Đà Nẵng**.



Khi đến **Đà Nẵng**, Doanh biết được có chuyến tàu đi từ **Đà Nẵng** đến **Quảng Ngãi** là gần với **My home** nhất nên Doanh quyết định chọn chuyến tàu này.



Sau khi mấy vài giờ đi tàu, cuối cùng Doanh cũng đã đến **Quảng Ngãi**, do **My home** ở gần ga tàu nên Doanh chỉ cần đi bộ là về được nhà mình.



Việc chia cách nút vào các tầng theo khoảng cách giải quyết được vấn đề khi đồ thị có nhiều nút.

2.4.6.2 Đánh giá

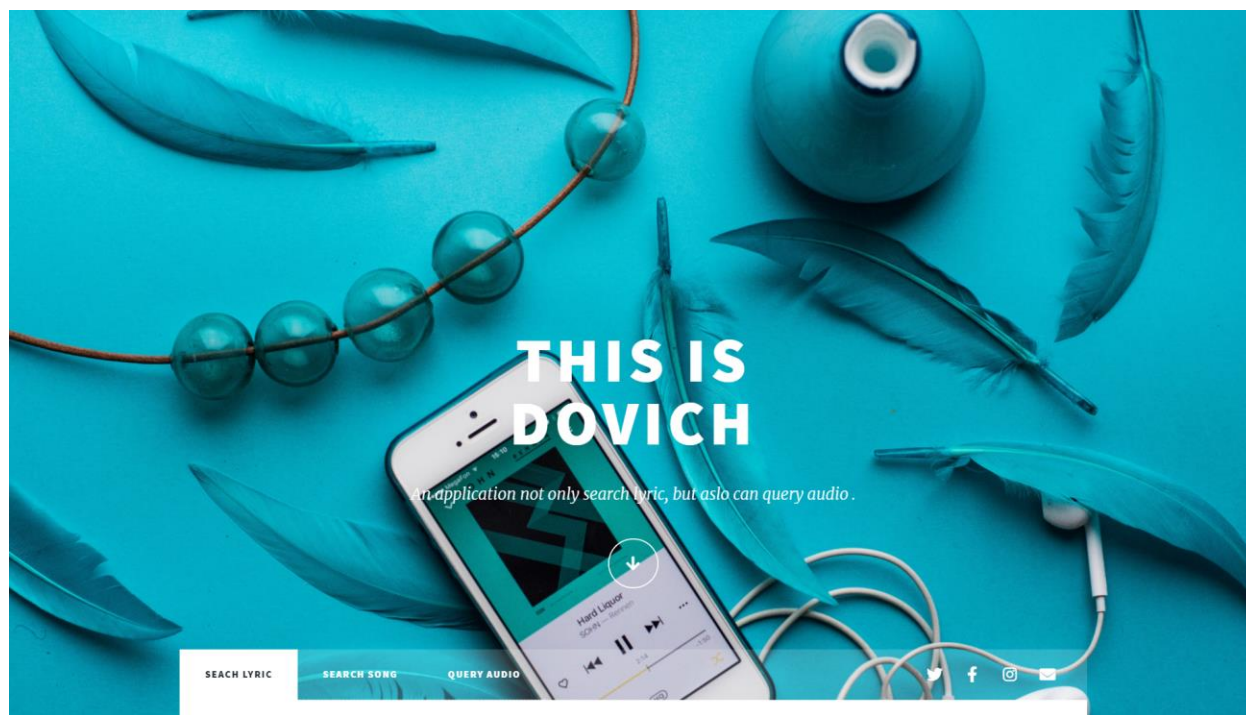
- Songs: chọn ngẫu nhiên 5 bài hát trong database, sau đó một đoạn nhỏ của mỗi bài ra để thực hiện phương pháp truy vấn.
- Number Query: 1 query ứng với đoạn âm thanh 20ms, cho nên Number Query sẽ tương ứng với số giây của đoạn âm thanh mà nhóm đã cắt ra từ 1 bài hát.
- Recall: là trung bình tỉ lệ kết quả đúng của truy vấn chia cho kết quả đúng thực sự của tất cả query trong một đoạn nhạc được cắt từ bài hát.

Songs	Number Query	Recall
Ăn - Hannah Hoang	203	0.87
Exile - Taylor Swift	247	0.56
Đò Tình Đã Lỡ - Dư Anh	196	0.73
Feel Me – Kage	271	0.65
Em Là Châu Báu - MCK	305	0.79

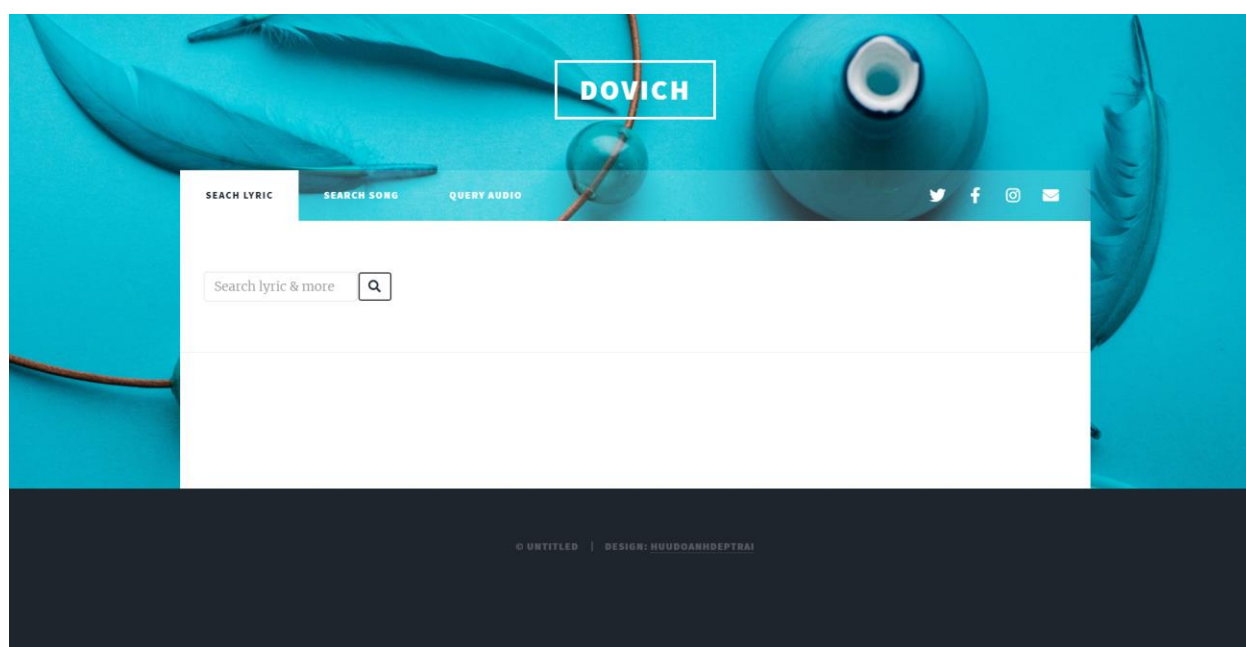
PHẦN 3: GIAO DIỆN HỆ THỐNG VÀ DEMO

3.1 Giao diện hệ thống

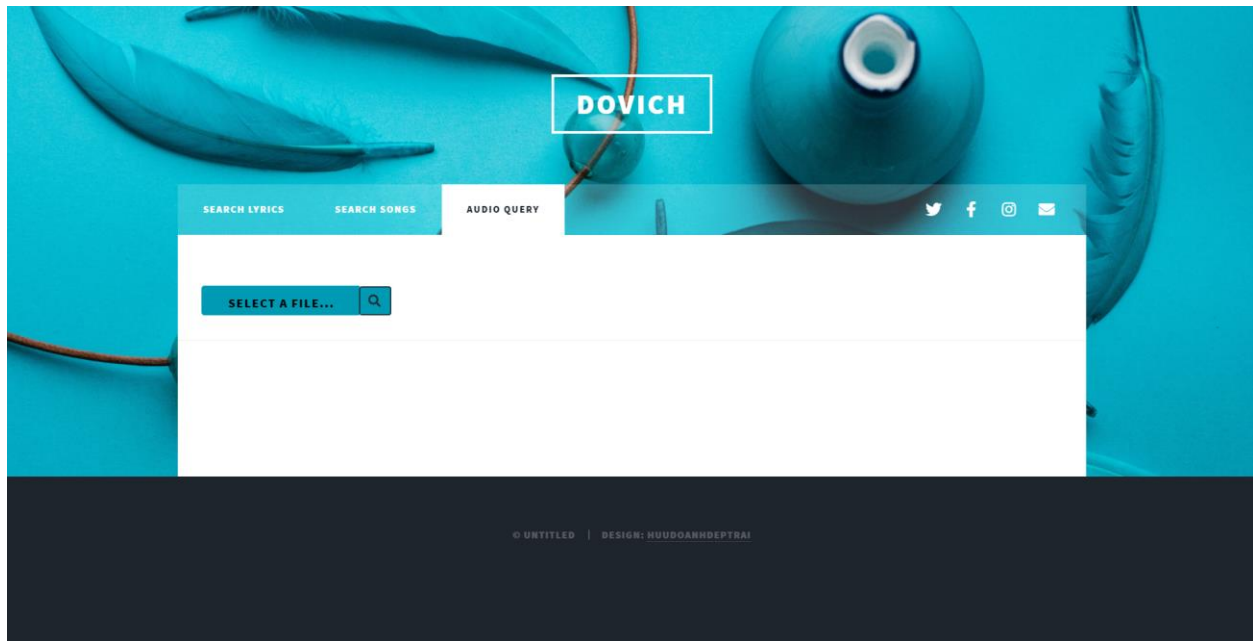
Một số hình ảnh về giao diện



Trang chủ của hệ thống



Trang tìm bài hát theo đoạn lời

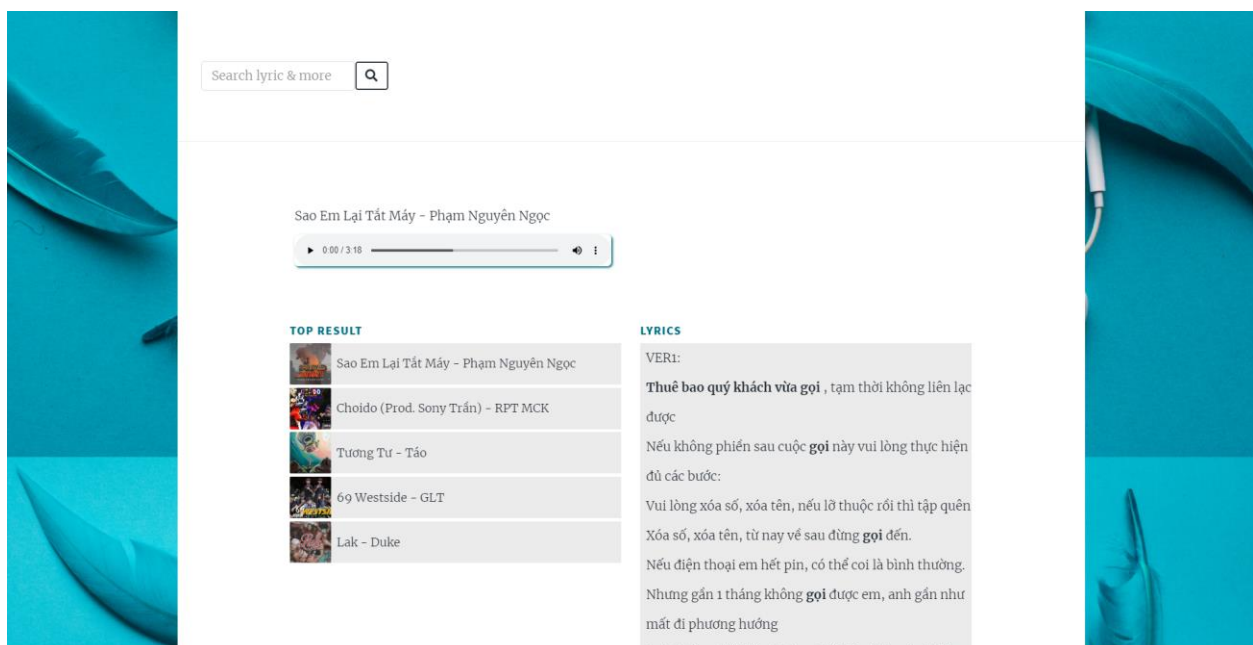


Trang tìm bài hát theo giai điệu

3.2 Demo

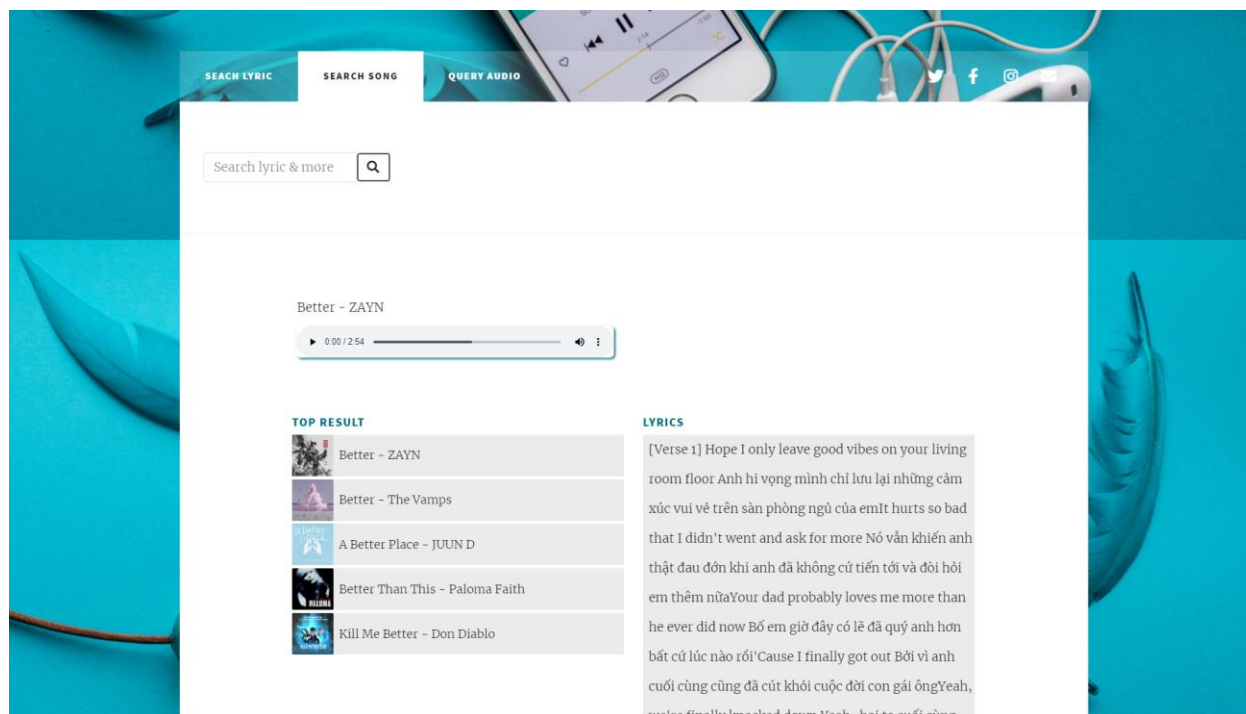
3.2.1 Chức năng Search lyric

Khi chúng ta tìm kiếm với câu là “Thuê bao quý khách vừa gọi” thì hệ thống sẽ trả về tên, cả đoạn nhạc của bài hát đó. Ngoài ra nó còn trả về các kết quả có lời gần giống với câu truy vấn mà chúng ta muốn tìm kiếm và lời của bài hát có xếp hạng cao nhất.



3.2.2 Chức năng Search song

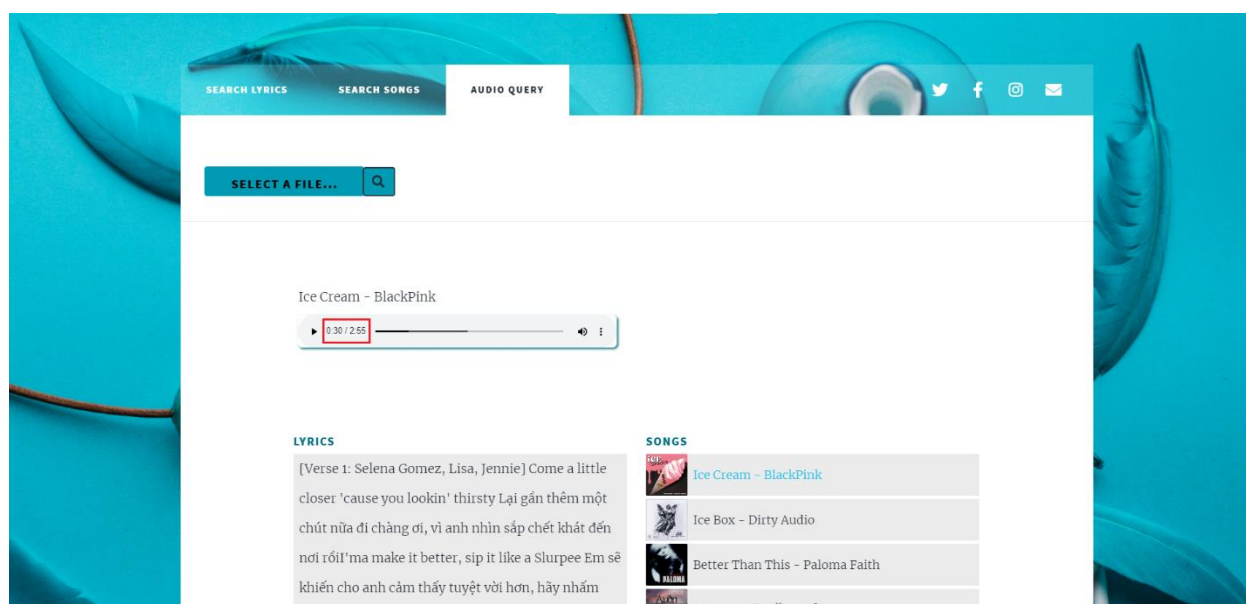
Về chức năng nó tương tự như phân giao diện search lyric



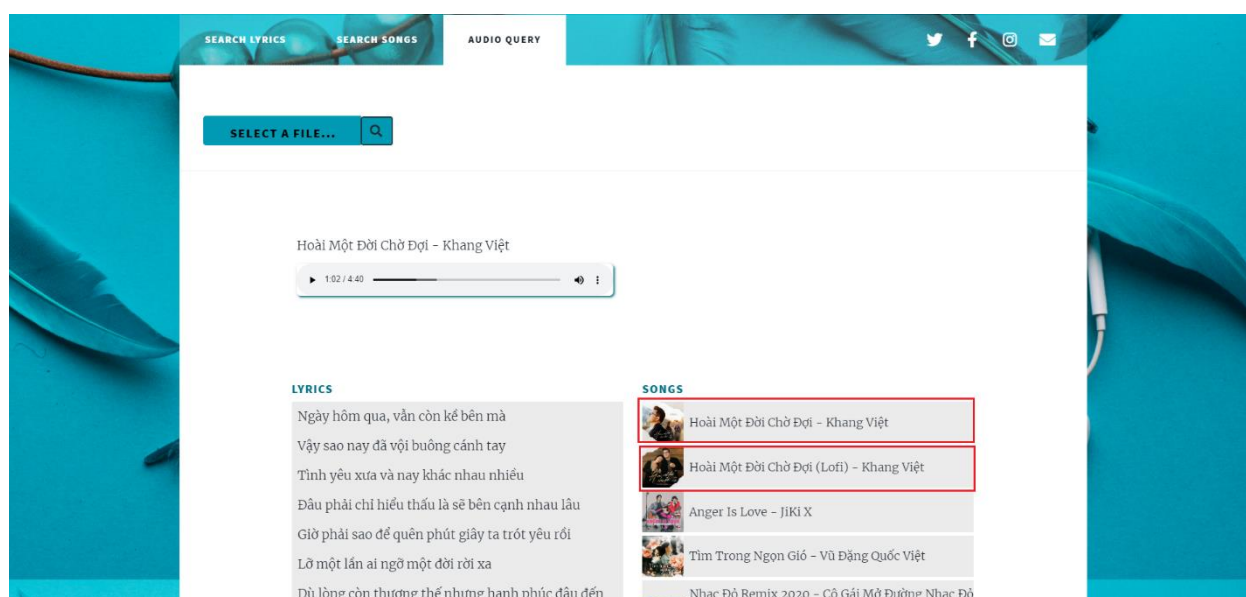
3.2.3 Chức năng Query audio

Ngoài các chức năng tìm kiếm tương tự như Search song, Search lyric thì truy vấn âm thanh còn có các chức năng như sau:

Có thể biết số giây của đoạn cắt nằm ở giây thứ mấy của bài hát. Ví dụ như đoạn cắt của bài hát “Ice Cream – BlackPink” nó bắt đầu từ giây thứ 30 và kết thúc ở giây thứ 60 thì kết quả xuất ra sẽ hiển thị giây đầu tiên ngay ở nơi bạn muốn nghe nhạc.



Hệ thống còn có thể nhận diện được các bài có cùng ca sĩ, hay các bài cover, remix, các bản phối khác nhau...



References

1. Malkov, Y.A., Yashunin, D.A.: *Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs*. *ArXiv e-prints* (Mar 2016)
2. Martin Aumüller, Erik Bernhardsson, and Alexander Faithfull (2018). *ANN-Benchmarks: A Benchmarking Tool for Approximate Nearest Neighbor Algorithms*.
3. Wen Li, Ying Zhang, Yifang Sun, Wei Wang, Wenjie Zhang, Xuemin Lin (2016). *Approximate Nearest Neighbor Search on High Dimensional Data — Experiments, Analyses and Improvement*.
4. Bernhardsson, E.: Annoy, <https://github.com/spotify/annoy>
5. Malkov, Y., Ponomarenko, A., Logvinov, A., Krylov, V.: *Approximate nearest neighbor algorithm based on navigable small world graphs*. *Inf. Syst.* 45, 61–68 (2014)
6. Yurymalkov, E.: Hnswlib, <https://github.com/nmslib/hnswlib>