ANATOMY ARCHITECTURE

VNSE SEARCH ENGINE

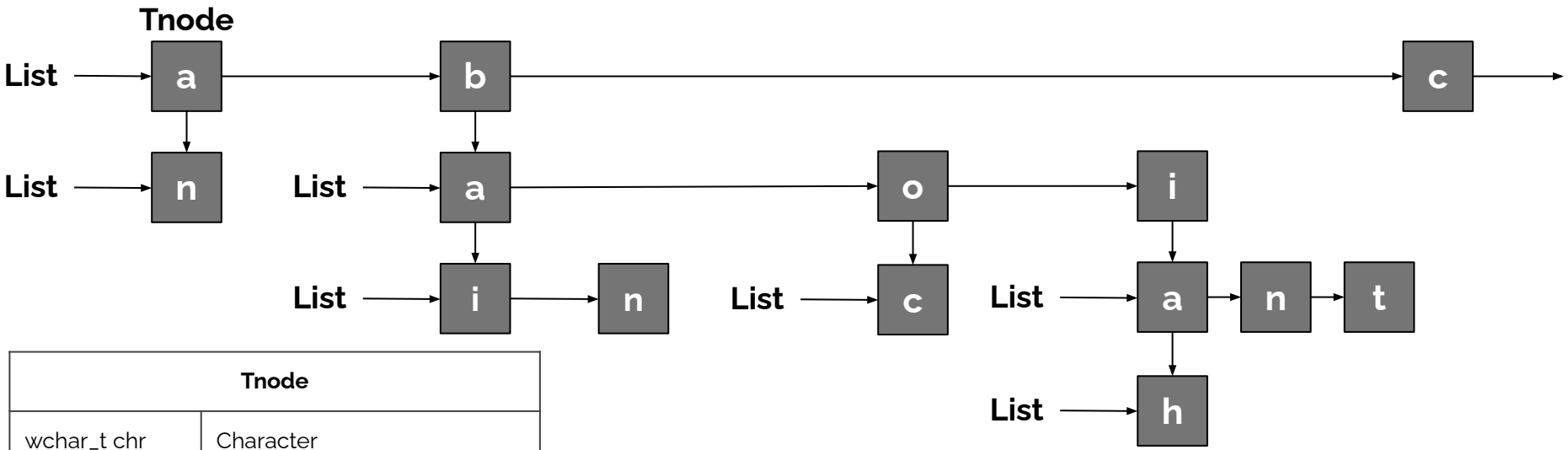
Lê Quang Trí - 20120022
Nguyễn Kông Đại - 20120448

Terminology

- Corpus: a resource that consists of all the text files.
- Document/Doc: a text file.
- Query: the query entered by users.
- Token: a word which is splitted in a sentence **by space,** also known as syllable in Vietnamese.
- Word: similar to token.
- Term: a meaningful word in Vietnamese. It can be a combination of multiple tokens/word/syllables.
- Lexicon: a resource that consists of word.
- Indexing: a stage to create metadata.
- Searching: a stage to find relevant documents for a query.

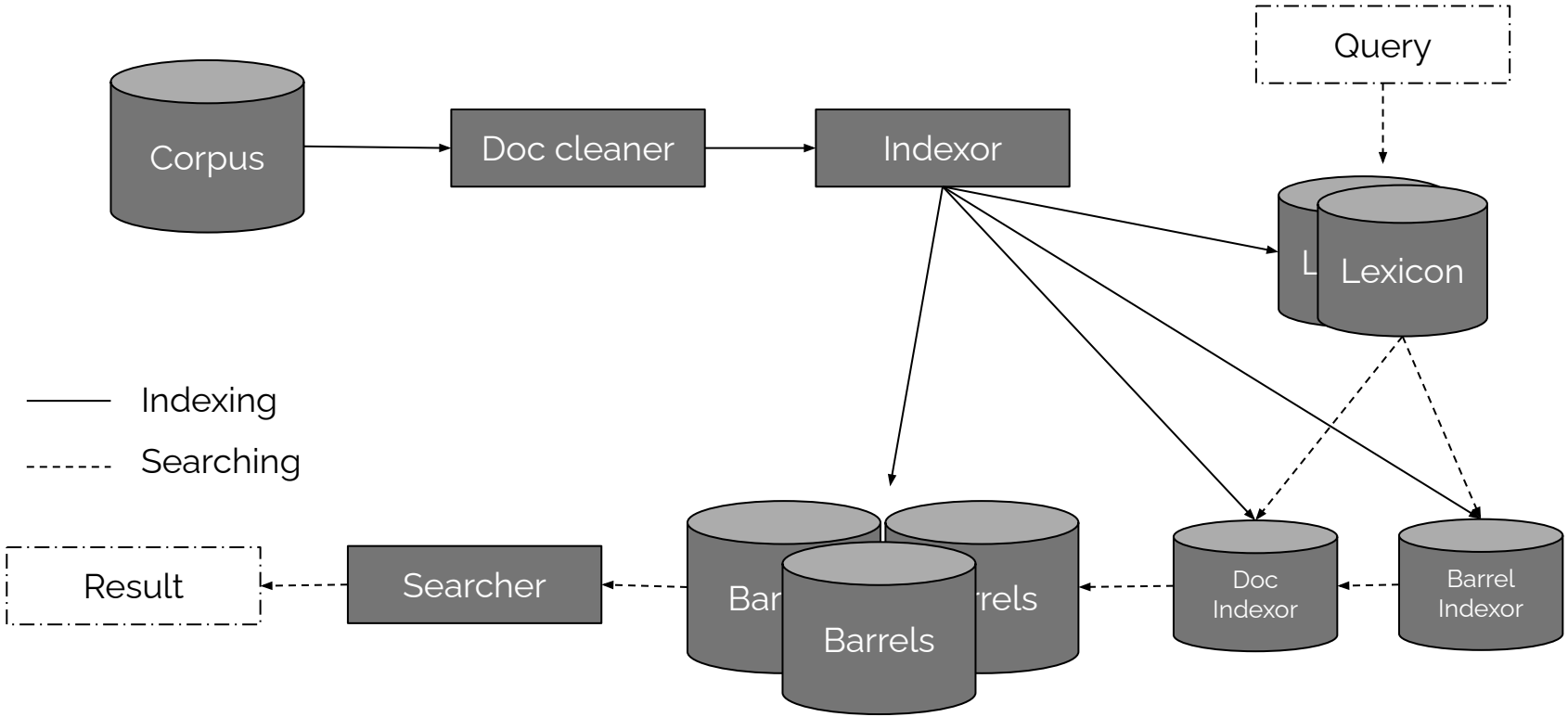
Lexicon Structure

Super Linked List

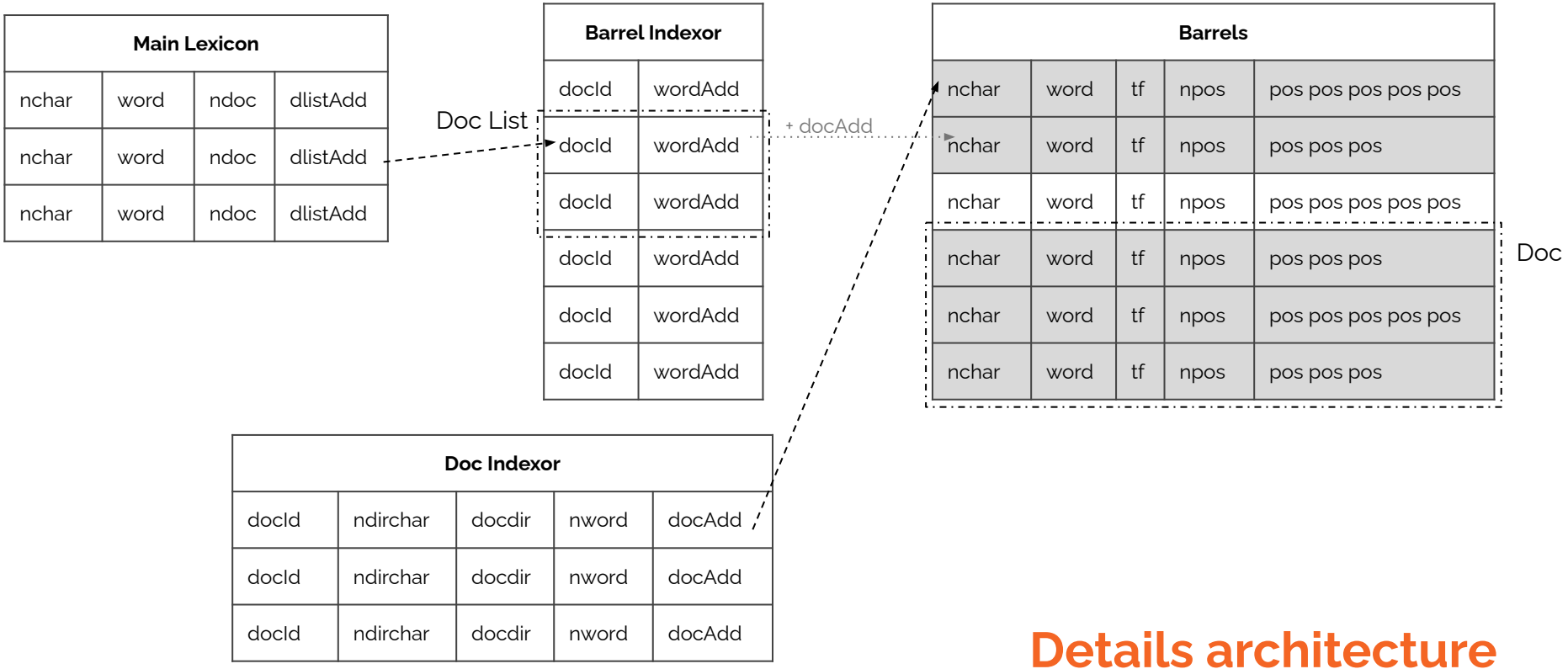


Tnode	
wchar_t chr	Character
SList branch	Branch
bool isWord	True if it is the end of a word
void* adapter	Other information

Search Engine Architecture



General Architecture



Details architecture

Indexing process

1. Read and clean document

Raw text

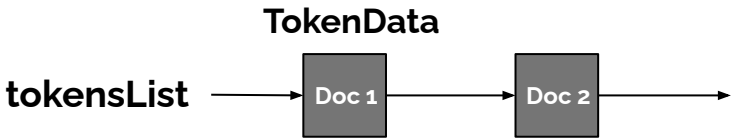
sự phối hợp hoàn hảo giữa máy nghe, xem nhạc kỹ thuật số và ổ cứng di động với dung lượng lên đến 30 GB.
Máy có thể chứa 15 ngàn bài hát và hơn 1.000 ảnh kỹ thuật số. Zen Vision M có thể phát các tập tin nhạc MP3/WMA/WAV/ MPEG với 5 chế độ chỉnh nhạc gồm normal, pop, rock, jazz, classic; xem được các file hình ảnh với định dạng JPEF, BMP, GIF, TIFF.

Clean text

sự phối hợp hoàn hảo giữa máy nghe xem nhạc kỹ thuật số và ổ cứng di động với dung lượng lên đến 30 gb máy có thể chứa 15 ngàn bài hát và hơn 1 000 ảnh kỹ thuật số zen vision m có thể phát các tập tin nhạc mp3 wma wav mpeg với 5 chế độ chỉnh nhạc gồm normal pop rock jazz classic xem được các file hình ảnh với định dạng jpef bmp gif tiff

Input: FILE* - Output: wchar_t*
Read a file and remove all redundant things like punctuation, unnecessary spaces,... convert capital letters to normal ones. Return a string (wchar_t*)

2. Make Word Array for a doc



TokenData	
wchar_t* word	Word
short position	Position of that token in text

Input: wchar_t* text - Output: SList tokensList
Make a tokens List from wchar_t* text and remove all stop words.

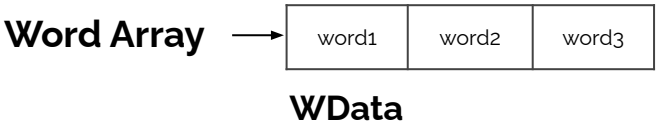
Example

đại nam việt nam không ở đại dương

Tokens list →

đại 0	nam 1	việt 2	không 3	ở 4	dương 5

2. Make Word Array for a doc

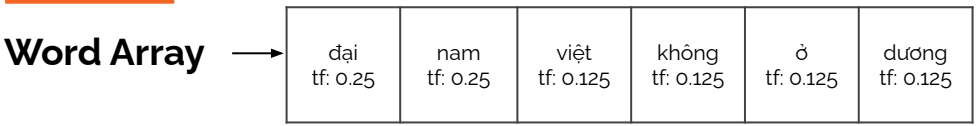


WData (Word Data)	
wchar_t* word	Word
float tf	Term frequency
short npos	Number of positions
short* posarray	An array of positions of that word in the doc

Input: tokensList - Output: Word array
Count the occurrences of each word in the document, calculate the term frequency and append it to word array.

Example

đại nam việt nam không ở đại dương



3. Write Barrels & Doc Indexor

Address Array → | word1 | word2 | word3 |

WAData

*docAdd

Doc Indexor

docId	ndirchar	docdir	nword	docAdd
docId	ndirchar	docdir	nword	docAdd
docId	ndirchar	docdir	nword	docAdd

WAData (Word Address Data)

wchar_t* word	Word
int wordAdd	Address to word's info in the doc. Must plus with docAdd to get the address to that word's info in barrel.

Example

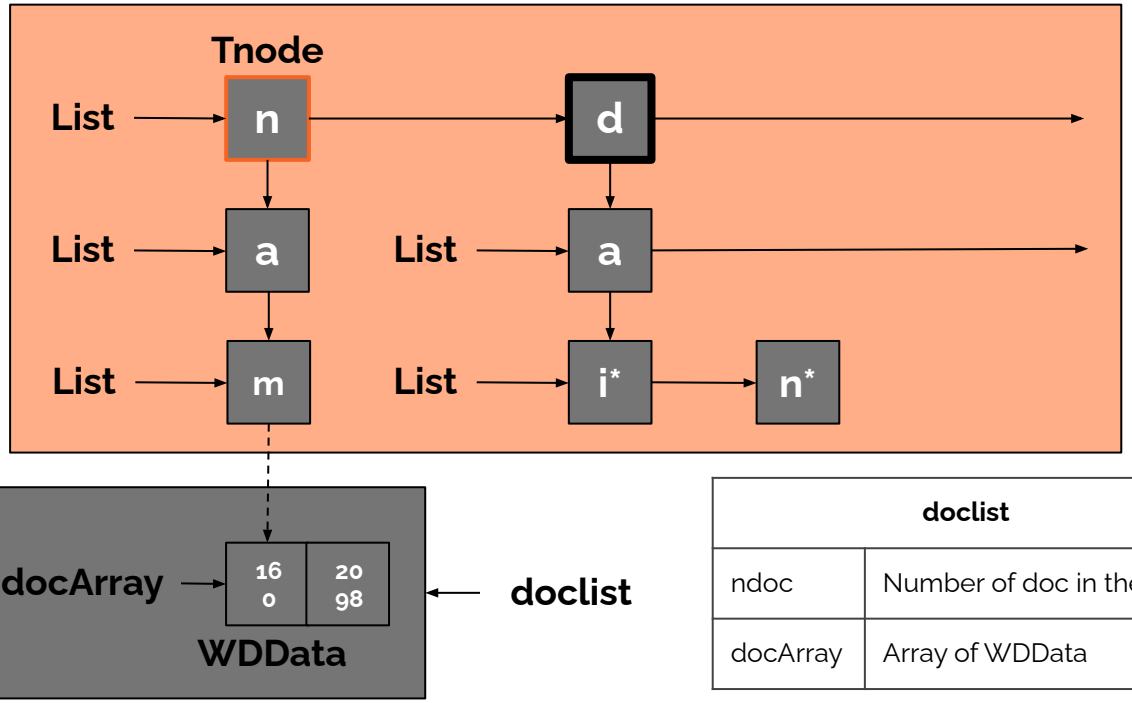
đại 0	nam 4	việt 8	không 12	ở 16	dương 20

Big Barrels

nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos

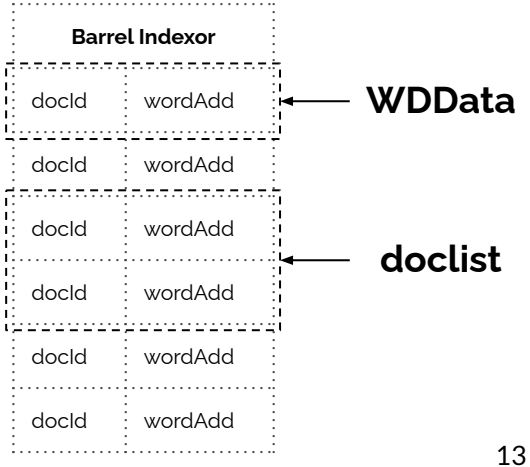
Input: Word Array, Barrels FILE - Output: Address Array
Write word array to barrels. Get the address of the first word to be docAdd of that doc. For every word, subtract the address by docAdd to get wordAdd.

4. Write Lexicon & Barrel Indexor - Method 1



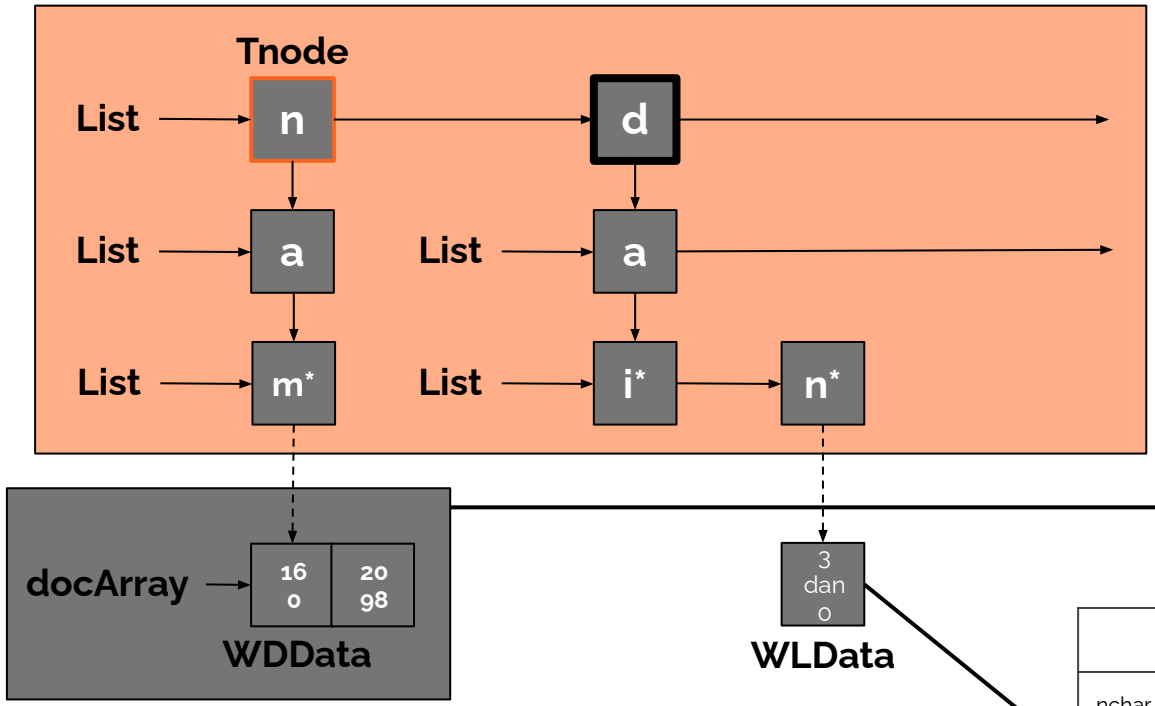
WDData (Word in Doc Data)	
docId	Id of the doc
wordAdd	Address to word's info in the doc. Must plus with docAdd to get the address to that word's info in barrel.

doclist	
ndoc	Number of doc in the list
docArray	Array of WDData



Input: Address Array - Output: Updated lexicon
Adapt word address in address array to lexicon.
Adapter of lexicon here are doclists.

4. Write Lexicon & Barrel Indexor - Method 1



WLData (Word in Lexicon Data)	
short nchar	Number of characters
wchar_t* word	Word
int ndoc	Number of docs
int dlistAdd	Address to doclist in barrel indexor

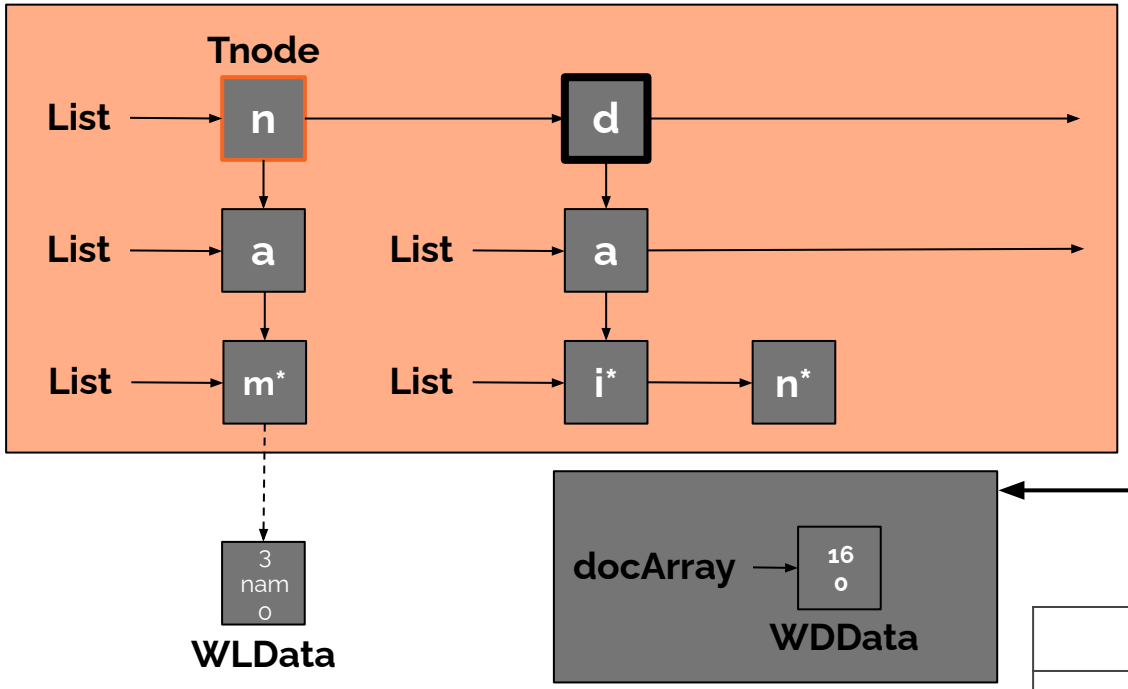
Barrel Indexor	
docId	wordAdd
docId	wordAdd
docId	wordAdd
docId	wordAdd
docId	wordAdd
docId	wordAdd

Main Lexicon			
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd

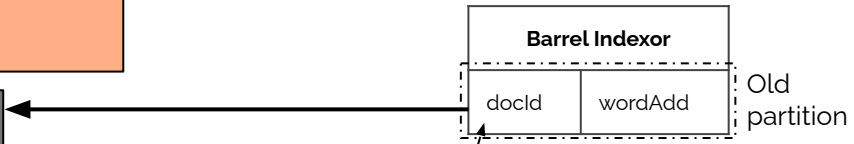
WLData

Input: Lexicon FILE, Barrel Indexor FILE - Output: Updated Lexicon
Save doclist to barrel indexor. Replace adapter of that word from doclist to WLData. Then, save all WLData to lexicon file.

4. Write Lexicon & Barrel Indexor - Method 2



WLData (Word in Lexicon Data)	
short nchar	Number of characters
wchar_t* word	Word
int ndoc	Number of docs
int dlistAdd	Address to doclist in barrel indexor



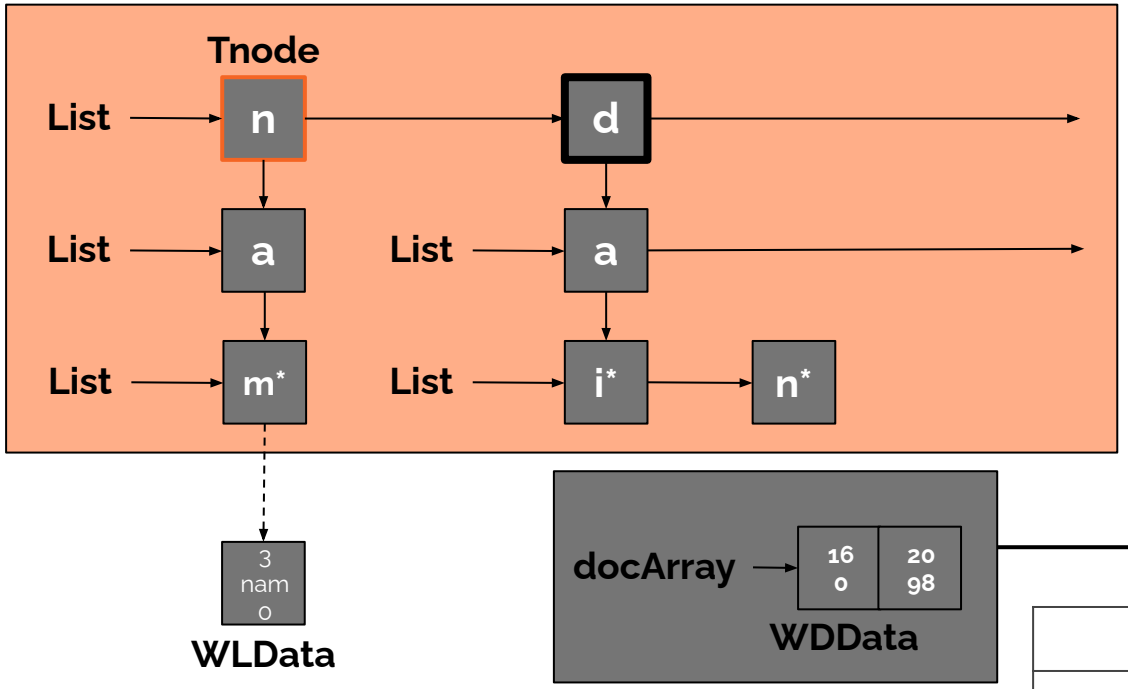
Input: Lexicon, Barrel Indexor FILE - Output: DocList

Find the adapting word in lexicon. If it doesn't exist, add word to lexicon and write to Barrel Indexor. If it exist, get the DocList from Barrel Indexor based on dlistAdd of that word.

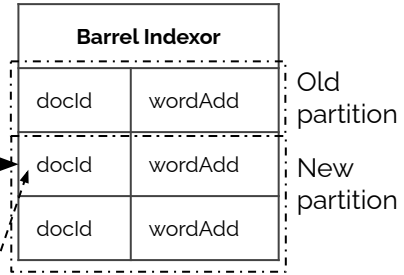
WLData →

Main Lexicon			
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd

4. Write Lexicon & Barrel Indexor - Method 2



WLData (Word in Lexicon Data)	
short nchar	Number of characters
wchar_t* word	Word
int ndoc	Number of docs
int dlistAdd	Address to doclist in barrel indexor

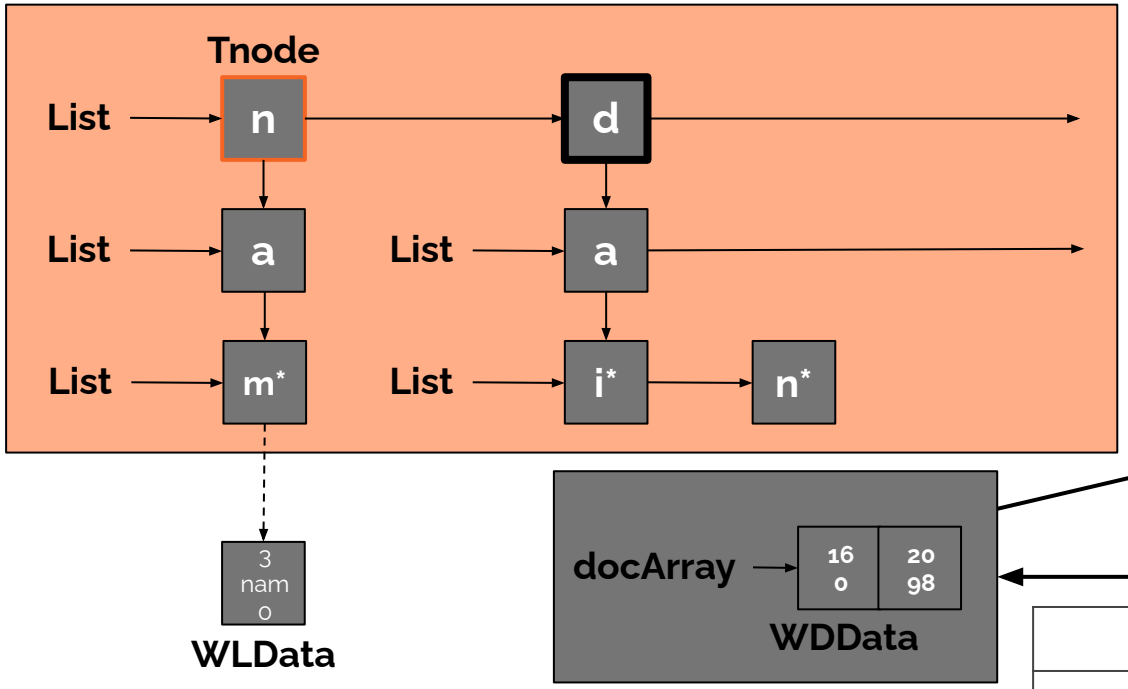


Main Lexicon			
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd

WLData

Input: Lexicon, Barrel Indexor FILE - Output: Updated Lexicon
Add new word info to DocList and write to the end of Barrel Indexor. Update the dlistAdd of that word in Lexicon.

4. Write Lexicon & Barrel Indexor - Method 2



WLData (Word in Lexicon Data)	
short nchar	Number of characters
wchar_t* word	Word
int ndoc	Number of docs
int dlistAdd	Address to doclist in barrel indexor

New data

Barrel Indexor	
docId	wordAdd
docId	wordAdd

Old data

Barrel Indexor	
docId	wordAdd
docId	wordAdd
docId	wordAdd

Main Lexicon			
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd

WLData

Input: Lexicon, Barrel Indexor FILE - Output: Updated Lexicon
For each word in lexicon get DocList from Barrel Indexor, write DocList to cache file and update dlistAdd. After all, rename cache file and remove old Barrel Indexor.

Searching process

1. Find word in Lexicon

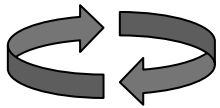
word word

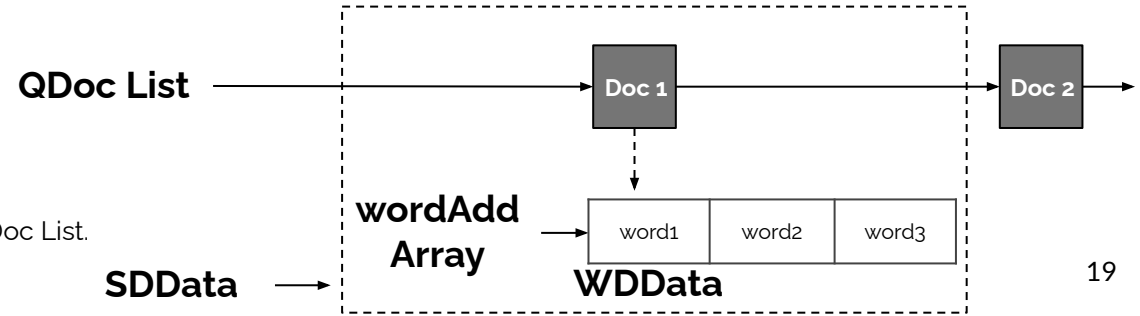
Query

Main Lexicon			
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd

Barrel Indexor	
docId	wordAdd
docId	wordAdd
docId	wordAdd
docId	wordAdd
docId	wordAdd
docId	wordAdd

SDData (Searched Doc Data)	
int docId	Id of the doc
int nword	Number of matched word
WDData* wordAddArray	Array of word address
float score	Score of that doc

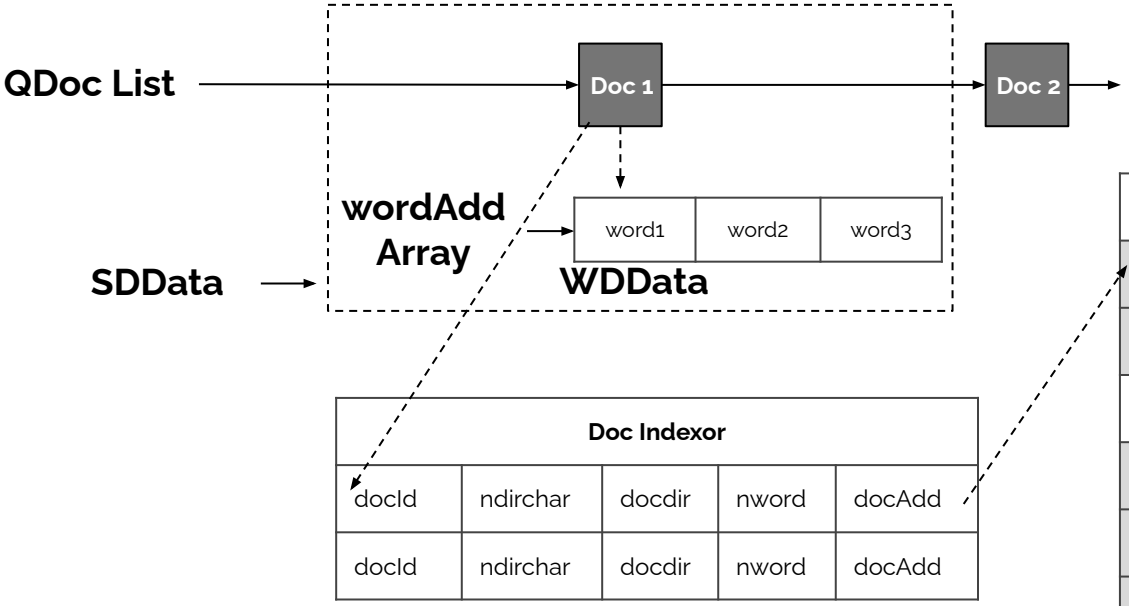




Input: wchar_t* Query - Output: QDoc List

For each word in query, find it in lexicon and get all corresponding doclist in barrel indexor. Then add to QDoc List.

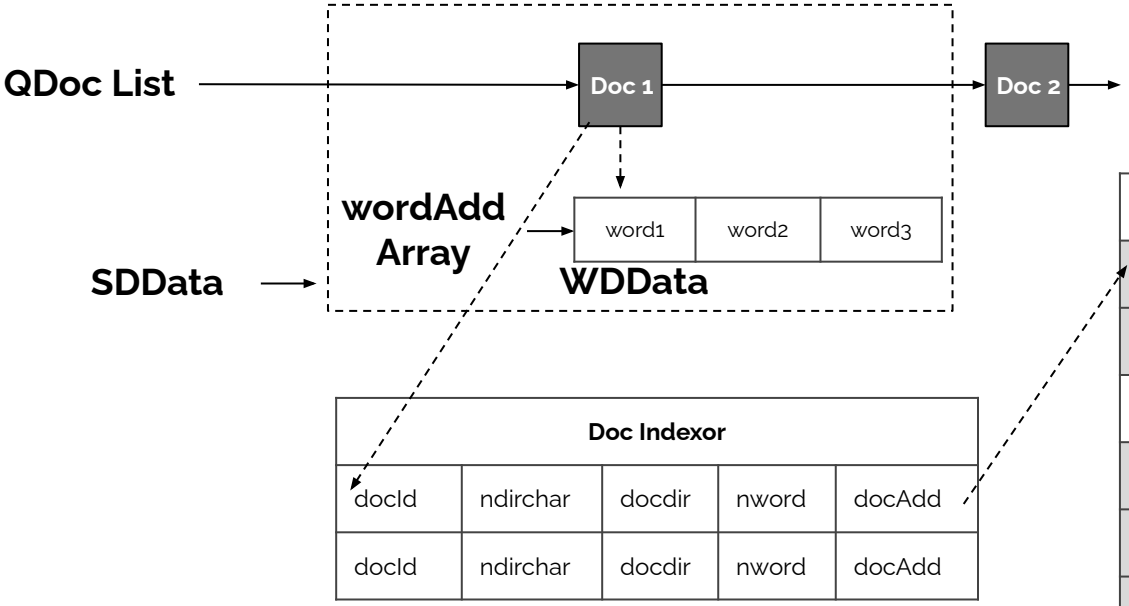
2. Get word data in Barrels



Barrels				
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos

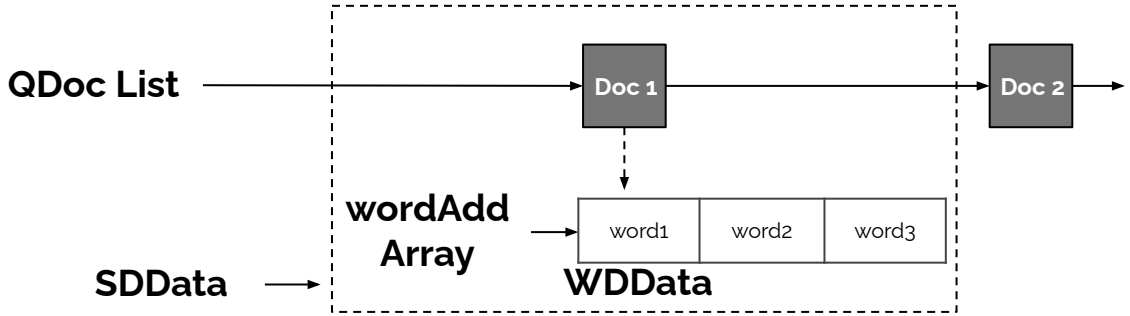
Input: QDoc List - Output: Word Data
For each document, find in doc indexor by docId to get docAdd. For each word in wordAddArray, get wordAdd + docAdd to get access to word data in barrels.

3. Compute score for document



Input: QDoc List - Output: Updated QDoc List
Compute the score for each document in QDoc List.

4. Sort document by score



Input: QDoc List - Output: Updated QDoc List
Sort the document by score and return the result.

Adding document process

1. Make word array for document

Main Lexicon			
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd

Barrel Indexor	
docId	wordAdd
docId	wordAdd
docId	wordAdd

Barrels				
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos

Doc Indexor				
docId	ndirchar	docdir	nword	docAdd
docId	ndirchar	docdir	nword	docAdd

Input: tokenList - Output: Word Array

Make word array for new document in the same way as indexing.

2. Save to Barrels & Doc Indexor

Main Lexicon			
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd

Barrel Indexor	
docId	wordAdd
docId	wordAdd
docId	wordAdd

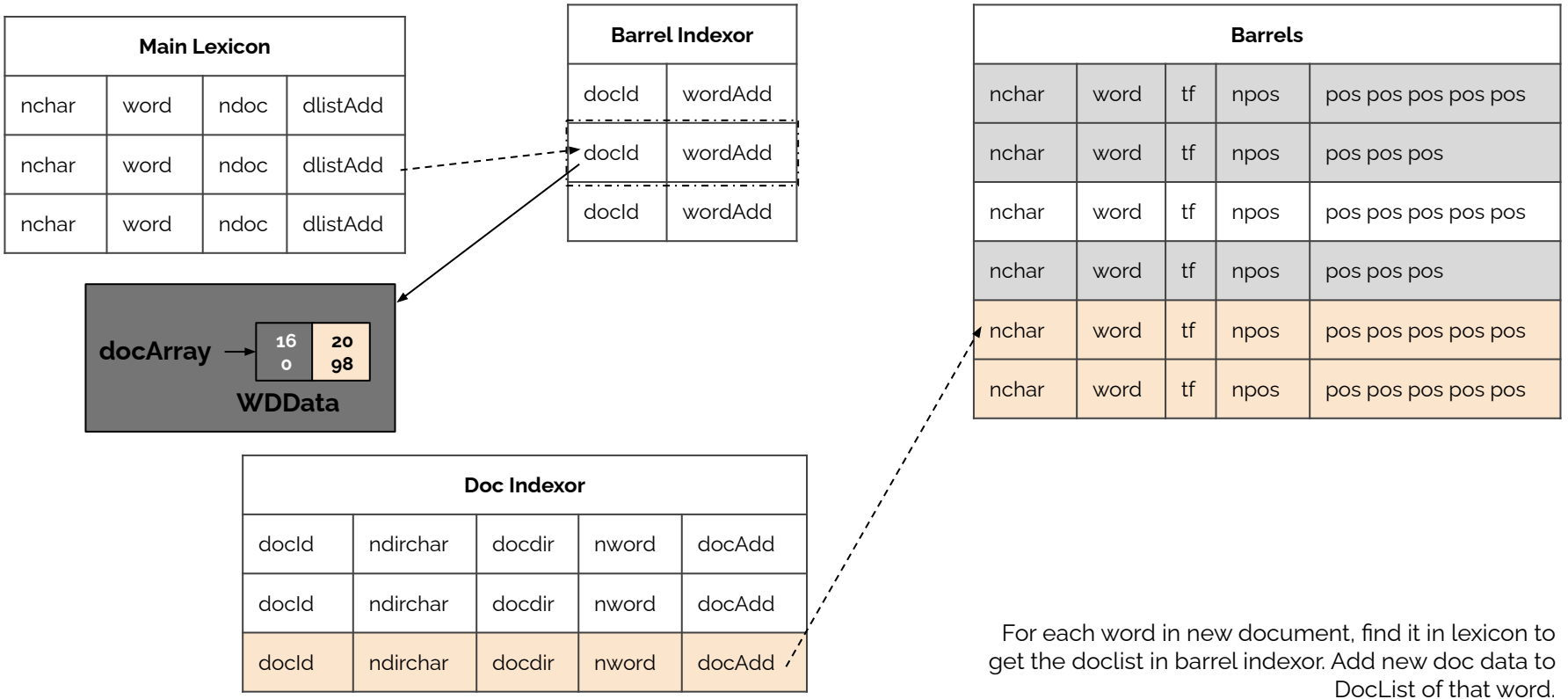
Barrels				
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos pos pos

New data

Doc Indexor				
docId	ndirchar	docdir	nword	docAdd
docId	ndirchar	docdir	nword	docAdd
docId	ndirchar	docdir	nword	docAdd

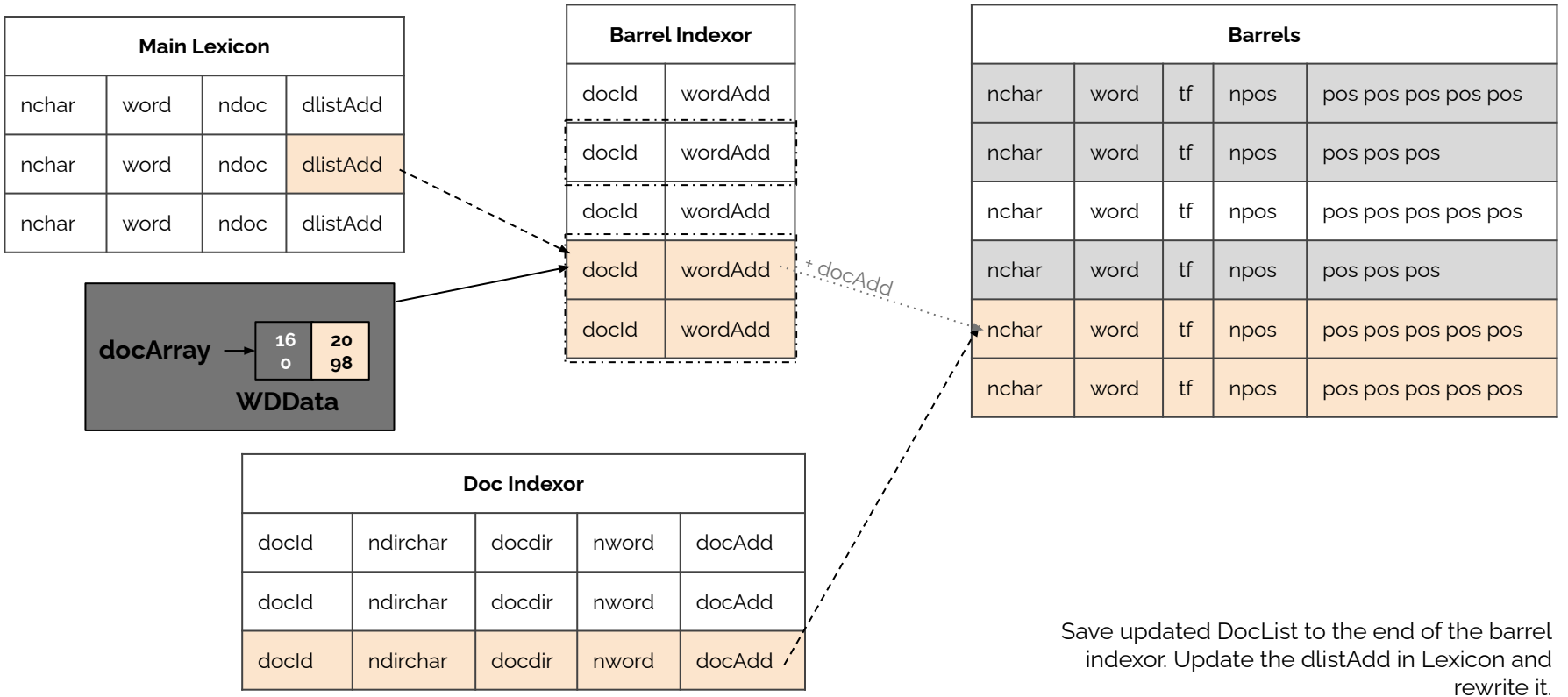
Input: Word Array - Output: Address Array
Save word array to the end of barrels and get address array in the same way as indexing.

3. Update DocList



For each word in new document, find it in lexicon to get the doclist in barrel indexor. Add new doc data to DocList of that word.

4. Save Barrel Indexor & Lexicon



Save updated DocList to the end of the barrel indexor. Update the dlistAdd in Lexicon and rewrite it.

Removing document process

1. Rewrite Doc Indexor

Main Lexicon			
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd
nchar	word	ndoc	dlistAdd

Barrel Indexor	
docId	wordAdd
docId	wordAdd
docId	wordAdd

Barrels				
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos pos pos

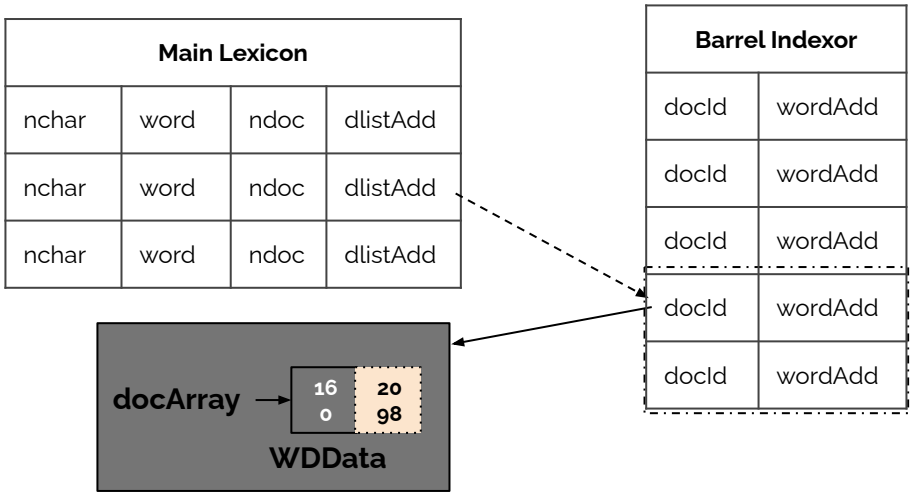
Doc Indexor				
docId	ndirchar	docdir	nword	docAdd
docId	ndirchar	docdir	nword	docAdd
docId	ndirchar	docdir	nword	docAdd

Old data

Input: doc directory - Output: Word Array

Find document in Doc Indexor and get docAdd to access the word array in Barrels. Rewrite the doc indexor to remove document.

2. Update DocList



Barrels

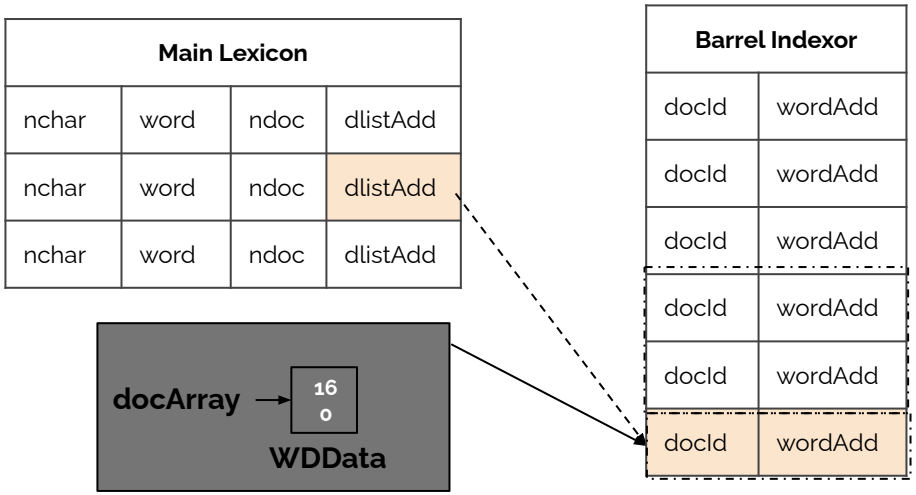
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos pos pos

Doc Indexor

docId	ndirchar	docdir	nword	docAdd
docId	ndirchar	docdir	nword	docAdd

For each word in Word Array, find it in lexicon to get the doclist in barrel indexor. Remove old doc data from DocList of that word.

3. Save Barrel Indexor & Lexicon



Doc Indexor

docId	ndirchar	docdir	nword	docAdd
docId	ndirchar	docdir	nword	docAdd

Barrels

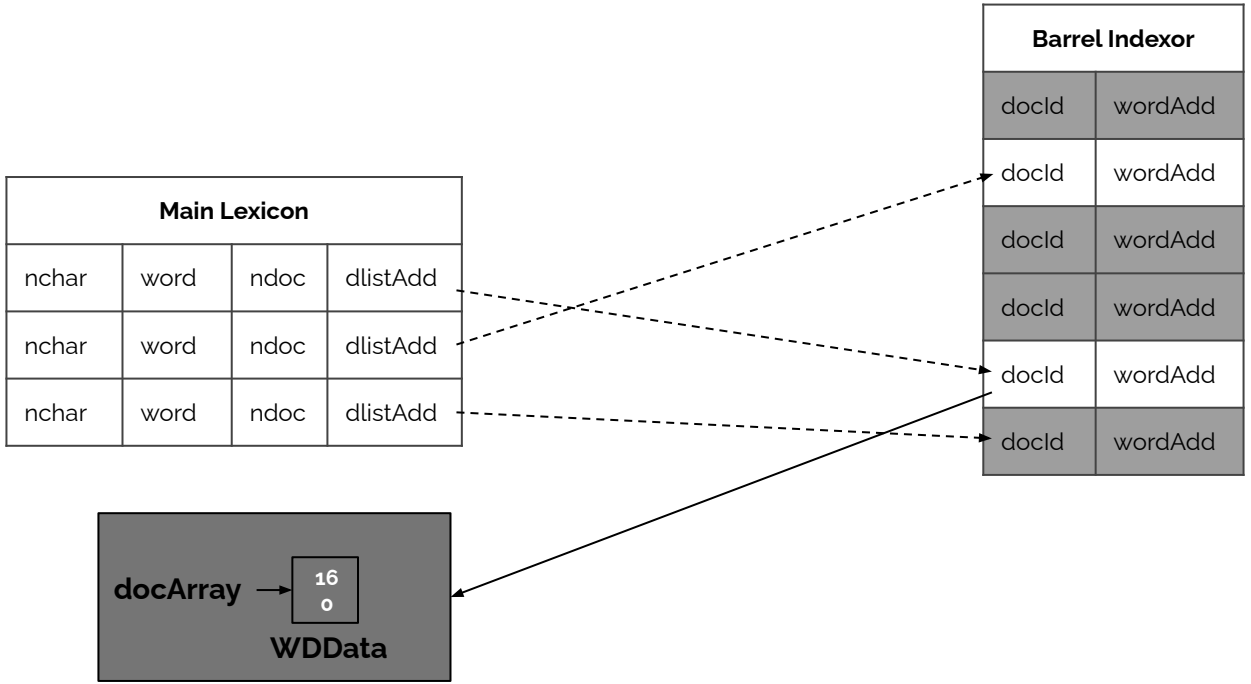
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos
nchar	word	tf	npos	pos pos pos pos pos
nchar	word	tf	npos	pos pos pos pos pos

Save updated DocList to the end of the barrel indexor.
Update the dlistAdd in Lexicon and rewrite it.

Rewrite techniques

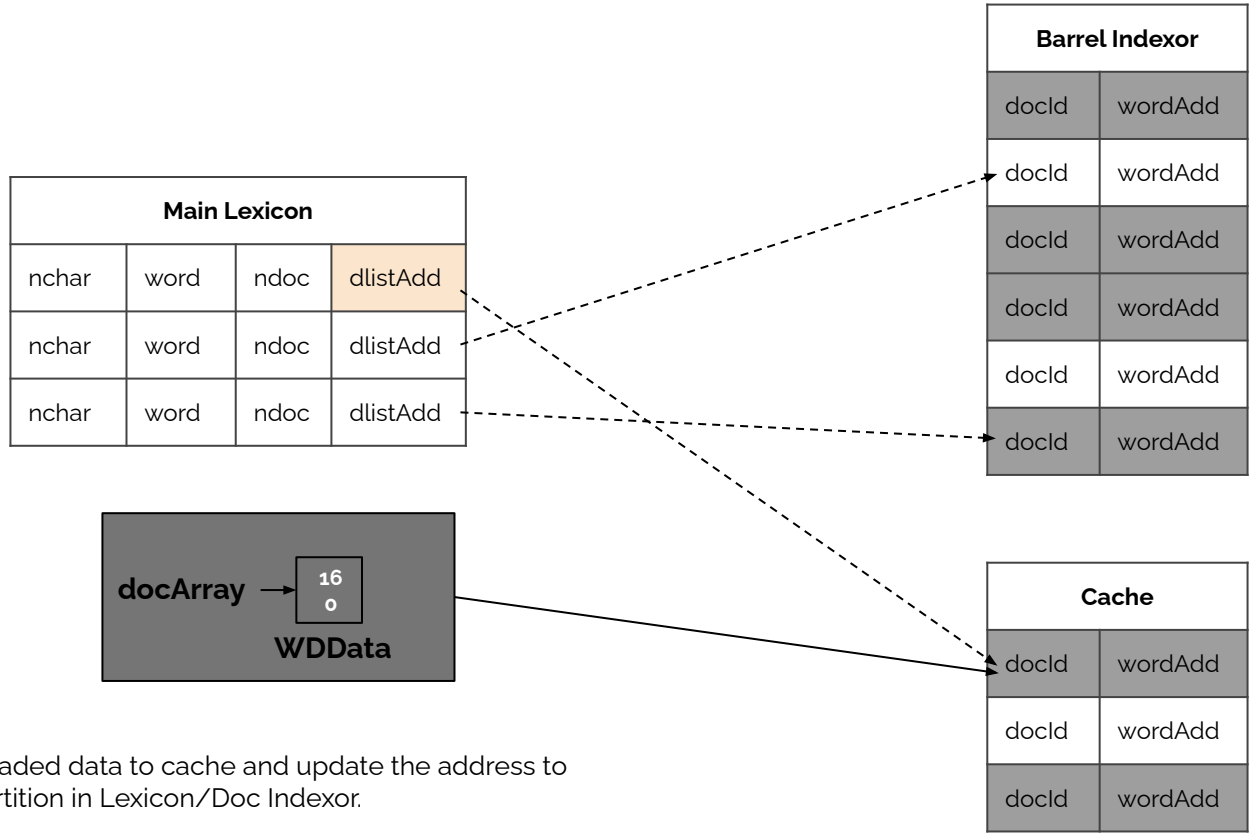
for Barrels & Barrel Indexor

1. Load data to RAM



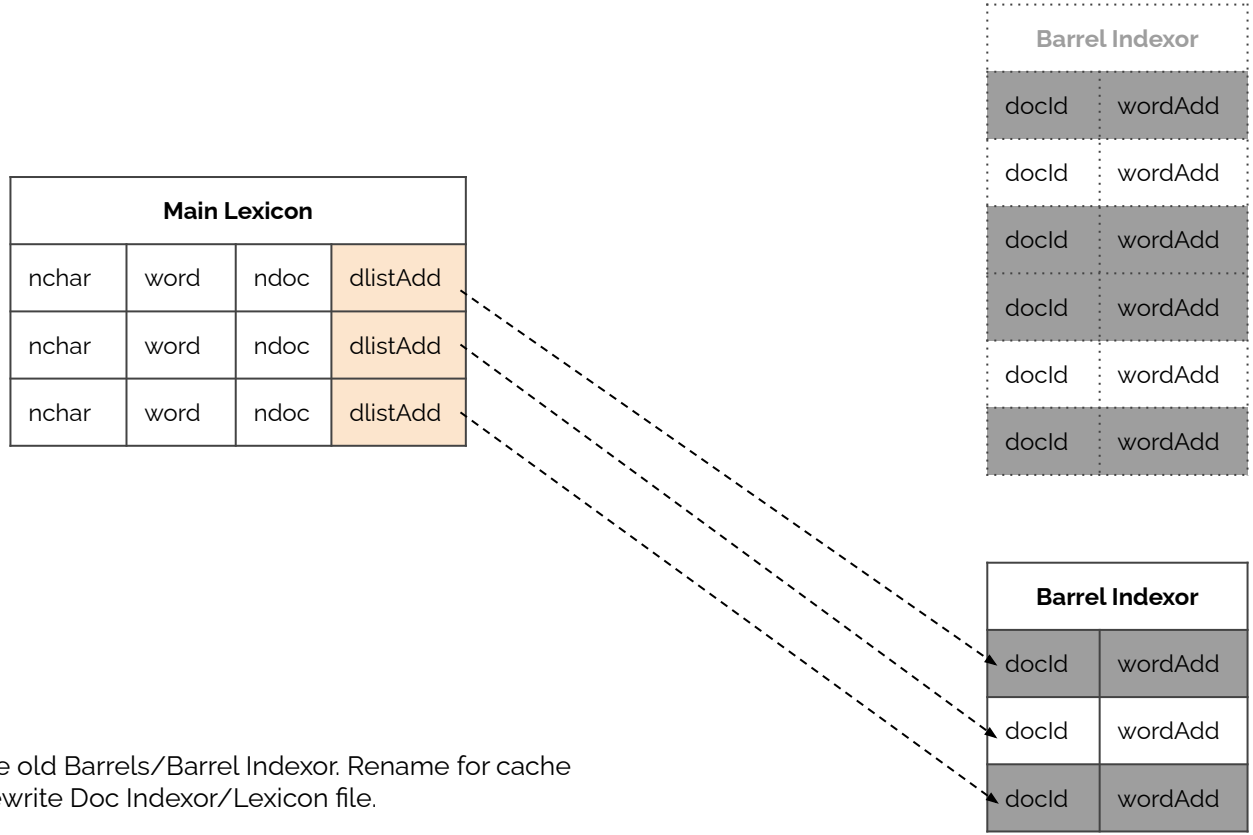
For each data in Lexicon/Doc Indexor. Load the corresponding data in Barrel Indexor/Barrels to RAM

2. Save data to cache



Save loaded data to cache and update the address to that partition in Lexicon/Doc Indexor.

3. Remove old file & rename cache file



Remove old Barrels/Barrel Indexor. Rename for cache files. Rewrite Doc Indexor/Lexicon file.

VNSE Search Engine

Algorithms

Please read in full research!

VNSE Search Engine

End.