

1. LSTM：具有記憶功能的 Neural Network，可以考慮到 input sequence 的順序 (order)問題，當中存在一個 memory cell 負責記錄前一層/前一次/之前的 data，memory cell 中的記錄是可以累加的，儲存的 data 會影響到 network 之後的 output。

LSTM 存在 input, output, forget 三個 gate 分別控制每一次的 input, output 是否需要被讀取/輸出，從而決定要讀取什麼 input，要取出什麼的 output 作為 final result。forget gate 控制 memory cell 是否要記錄這次的 data 或是否要清空，從而控制每次 output 的結果。

2. E step：假設未知的 data 為  $z$ ，第  $k$  個 iteration 的參數 set 為 0，已知的 data 為  $x$ 。先假設  $z$  有很多不同的可能值，且不同的值有不同的機率，利用  $x$  及 0 可以求出  $z$  的一個 distribution，最後則可求出 objection function 對於這個 distribution 的期望值。

M step：通過 Maximize objection function 的期望值，generating 一組最好的參數作為下一個 iteration 的參數使用。

經過一定的 iteration 後就會收斂。

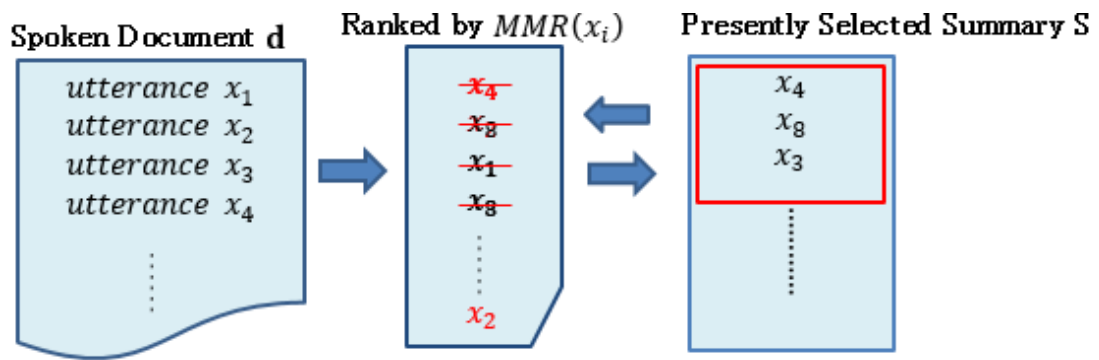
3. 3 key elements：

1. Speech recognition and understanding：指機器利用 user 第  $n$  句話及之前到  $n-1$  句話為止對 user 的了解，來對 user 第  $n$  句話進行解讀。
2. Discourse analysis：機器利用對 user 第  $n$  句話的解讀及之前到  $n-1$  句話為止對 user 的了解來增加機器對 user 的了解(在  $n-1$  句時對 user 的了解變成在第  $n$  句話時對 user 的了解)。
3. Dialogue management：利用機器到第  $n$  句話時對 user 的了解，來決定出下一個 action。

4. MMR：從 document 中選出與 document 內容相似的句子，用這些句子生出 document 的 summary。

MMR 有兩個步驟，relevant：算出 spoken/document 中每一句與全文的相似度，排序後則可以選出一些與文章最相關的句子作 summary 用。Non-redundant：避免選出來的 summary 中出現意思上具有重複性的句子(太像的句子)，把可以選出當 summary 的句子與 summary 現有的句子計算相似度，只把不相似用作 summary 用，避免 summary 中出現冗餘的資訊。

Example：



5. WFST：以 Finite State Machine 的形式表示出一種 input 對應到 output 的所有可能路徑，對於當中每一個 state 都會有它的 input，output 及 weight。可以透過 WFST 將 input 轉換成 output，並為 output/input 根據 weight 來進行評分。例如可以將中文->英文 / 語音->文字 並評分。

在 speech recognition 上，WFST 可以由四個 WFST composition 組成。

1. HMM topology：把每一個 HMM state sequence 對應到一個 context-dependent phoneme，當中 transducer 的 weight 為 state->phoneme 的機率，透過這個機率可以找到最合適的 phoneme。
2. Context-dependency：把每一組 context-dependent phoneme(e.g. tri-phoneme) 轉成 context-independent phoneme(phoneme)
3. Lexicon：把一組 phoneme 轉換成該 phoneme 所代表的一個 word，weight 為 phoneme 對應到發音的機率，透過這個機率可以找到最有可能的代表 word。
4. N-gram model：每一個 word 是一個 state，每一個 input 進來的 word 會直接 output，同時得到該 word 與前面其他 word 連起來的機率，即可以得到 word 在某個順序時的分數。

透過這四個 transducer，可以求出一組 HMM states 所對應的 word，再配合一個 acceptor U，透過將聲音訊號的 frames 轉換成 HMM states，即可以把聲音訊號換成文字。

6. 在 Spoken Content Retrieval 中，可以先把聲音->文字，再用文字的搜尋系統找出 result，但只把聲音轉成一個文字結果，可能會有辨識 error 的問題。利用 lattices 可以考慮聲音->文字的所有可能結果，減少辨識 error 的問題，找出更好的 result。

Problem：考慮聲音->文字更多的可能性可以包含正確的 words，但也會包含更多的 noisy words/incorrect words，導致錯誤率也有可能提升。很多時候正確的 words 會是 OOV，並不包含在 lattices 裡，所以 lattices 並不能解決 OOV 問題。

Subword：subwords 指把每個 word 拆成比 word 更小的 unit 來表示，當 OOV

拆成 subwords 時，以 subwords 來在 lattices 裡找有機會可以找出一個包含 OOV subwords 的路徑，解決 OOV 問題。

7. PCA：對於多維度的 data，希望可以找出一個較低維度的 vector space，當令 data 投影到 vector space 時，可以令不同類型的 data 更分散，同時保有 data 中重要的資訊。在不損失 data 資訊的情況下，減少參數量，提升 model 的訓練效率。
8. SA：對於一個聲音辨識 model，user 一開始先使用通用的 model，隨後 model 會漸漸根據 user 使用時的聲音 data 來調整 model，可以針對不同的 user 來提升 model 準確率。  
MLLR：把 model 中的 Gaussian 分成不同 group，相似的 gaussian 放在同一 group，在調整 model 時，根據 group 內有 data 的 gaussian 得到一組參數，group 內所有的 gaussian 都用這一組參數來進行調整，所以沒有 data 的 gaussian 也可以調整。而每個 gaussian 是利用 Linear regression，透過求出 maximum likelihood 來作調整。
9. MF：利用 Matrix 來表示不同 user 對不同 document 的行為(e.g. 評分/點擊)，利用此資料系統可以推論出與 user 行為有關的因素(e.g. 運動/美容)，從而找出符合該因素的其他 document，達到推薦效果。
10. HEQ：用來調整 data 的 distribution。先把 data 的 distribution 作排序，然後根據 data 作出一個 probability density function 的 distribution，分別對兩個 distributions 作積分，把 data 的值依照積分的結果一一對照，就可以對應到 probability density function 的 distribution 中相應的值，達到改寫 distribution 的效果。
11. PRF：在 Retrieval model 中，training data 都不好取得/不多。PRF 透過自動產生 training data 來對 search result 作 re-rank 動作，來改善 result 結果。  
Step：對於一個 query 的 results，假設排序後的前 N 個 result 是對的及後 M 個 result 是錯的，然後令前 N 及後 M 個以外的其他 result data 分別跟前 N 個及後 M 個 results 比較相似度。對於相似於前 N 個 result 的 data 提升它的排名，否則相似於後 M 個 result 的 data 降低它的排名，以此進行 re-rank 的動作，以 re-rank 的結果作為 Final Results 傳給 user。透過這樣的流程來改善搜尋的效果。