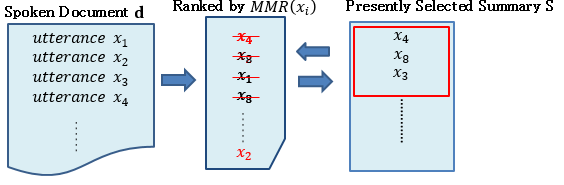
DSP Final Jun. 21, 2017

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spech 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-phone) 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。

1. 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。
2. 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。
3. 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gaussain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gaussain也可以調整。而每個gaussian是利用Linear regression，透過求出maximum likelihood來作調整。
4. 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。
5. 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。
6. 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。