

What is the fundamental frequency (F0) in acoustics? And why might it be useful for a spoken dialogue system to measure it?

- 1) the fundamental frequency F_0 is lowest frequency of the sound wave
- 2) it corresponds to the speed of vibration of the vocal folds
- 3) the fundamental frequency correlates with the pitch, which gives us the intonation of the speech signal. And intonation is important for a spoken dialogue system - for instance, a rising intonation at the end of a signal will indicate an interrogative utterance in many language

IR-based chatbots

Assume you wish to develop an IR-based chatbot. To make simple, you decide to use cosine similarity over TF-IDF-weighted vectors. You collect a small corpus of 10 utterances:

- 1 *Hi Charles!*
- 2 *Hello Elsa!*
- 3 *How are you?*
- 4 *Fine, and you?*
- 5 *A bit tired.*
- 6 *Why?*
- 7 *Busy at work.*
- 8 *But you are doing great!*
- 9 *Thanks.*
- 10 *See you later!*

- 1) Compute the TF-IDF values for the tokens in the utterances 3, 4 and 8. Don't forget to include the punctuation in your computations.
- 2) Now assume that you are given the following user input:

How are you doing?

What will be the answer selected by the chatbot trained on the corpus above? Describe your calculation steps.

utterance 3:

- "How": $\log(10/1) = 1$ (assuming a base 10)
- "are": $\log(10/2)$
- "you": $\log(10/4)$
- "?": $\log(10/3)$

utterance 4:

- "Fine": $\log(10/1)$
- ", ": $\log(10/1)$
- "and": $\log(10/1)$
- "you": $\log(10/4)$
- "?": $\log(10/3)$

utterance 8:

- "But": $\log(10/1)$
- "you": $\log(10/4)$
- "are": $\log(10/2)$
- "doing": $\log(10/1)$
- "great": $\log(10/1)$
- "!!": $\log(10/4)$

Note: I assume here a simple version of TF that simply express the number of times the token appears in the utterance (and since all tokens appear once, the TF is then 1). One can also rely on definition of TFs that divide the counts by the total number of tokens in the document.

Question 2: Response selection

The formula to use is the following

1. Return the response to the most similar turn: Given user query q and a conversational corpus C , find the turn t in C that is most similar to q (for example has the highest cosine with q) and return the *following* turn, i.e. the human response to t in C :

$$r = \text{response} \left(\underset{t \in C}{\operatorname{argmax}} \frac{q^T t}{\|q\| \|t\|} \right) \quad (26.1)$$

We start by looking at the IDF values for the query:

- "How": $\log(10/1)$
- "are": $\log(10/2)$
- "you": $\log(10/4)$
- "doing": $\log(10/1)$
- "?": $\log(10/3)$

The euclidian norm $\|q\|$

is $\sqrt{1^2 + (1 - \log 2)^2 + (1 - \log 4)^2 + 1^2 + (1 - \log 3)^2} = 1.709$ (although it doesn't really matter to search for the most relevant utterance)

And we can then compute the cosine similarity for the three sentences:

- utterance 3:
 - $q^T t = 1^2 + (1 - \log 2)^2 + (1 - \log 4)^2 + (1 - \log 3)^2 = 1.920$
 - $\|t\| = \sqrt{1^2 + (1 - \log 2)^2 + (1 - \log 4)^2 + (1 - \log 3)^2} = 1.386$
 - which means the cosine similarity is $\frac{1.920}{1.386 \cdot 1.709} = 0.811$
- utterance 4
 - $q^T t = (1 - \log 4)^2 + (1 - \log 3)^2 = 0.432$
 - $\|t\| = \sqrt{1^2 + 1^2 + 1^2 + (1 - \log 4)^2 + (1 - \log 3)^2} = 1.852$
 - which means the cosine similarity is $\frac{0.432}{1.852 \cdot 1.709} = 0.136$
- utterance 8:
 - $q^T t = (1 - \log 4)^2 + (1 - \log 2)^2 + 1^2 = 1.647$
 - $\|t\| = \sqrt{1^2 + (1 - \log 4)^2 + (1 - \log 2)^2 + 1^2 + 1^2 + (1 - \log 4)^2} = 1.950$
 - which means the cosine similarity is $\frac{1.647}{1.950 \cdot 1.709} = 0.494$

In other words, utterance 3 is the most similar to the query. Based on the mechanism described above, the chatbot will therefore select the utterance coming after utterance 3, which is "Fine, and you?".

1) What is a Markov Decision Process (MDP)? Give a formal definition.

2) How can MDPs be used in dialogue management?

Markov Decision Processes

- A MDP is defined as a tuple $\langle \mathbf{S}, \mathbf{A}, \mathbf{T}, \mathbf{R} \rangle$, where:
- $\mathbf{S}$ is the *state space* (space of possible states in the domain)
 - $\mathbf{A}$ is the *action space* (possible actions for the agent)
 - $\mathbf{T}$ is the *transition function*, defined as $T(s, a, s') = P(s'|s, a)$. It is the probability of arriving to state $\mathbf{s}'$ after executing action $\mathbf{a}$ in state $\mathbf{s}$.
 - $\mathbf{R}$ is the *reward function*, defined as $R : S \times A \rightarrow R$. It is a real number encoding the utility for the agent to perform action $\mathbf{a}$ while in state $\mathbf{s}$.

We can use MDP to represent dialogue policies, which are mappings between dialogue states to system actions. The idea is that:

- the state represent the current dialogue state (for instance as slot-values)
- the action space represent the possible actions of the dialogue system
- the transition function represent the dialogue dynamics
- and the rewards encode what we want the dialogue system to achieve, for instance satisfy a user request, minimise the number of turns, etc.

Given this formalisation, one can apply various reinforcement learning methods to automatically optimise dialogue policies from experience (which may be real dialogues or simulated ones).