



Each line transmitted will have the following structure:

Message Flag:

- F: Full version;
- C: concise version;
- O: other information

Sender:

Forward Path: (English to Persian):	Backward Path: (Persian to English):
A: ASR	a: ASR
M: MT	m: MT
T: TTS	t: TTS
D: Dialog Manager.	D: Dialog Manager (or 'd' if needed)
G: GUI	g: GUI

Receiver:

same as above

Content Flag:

- I: ID of the speech;
- T: text;
- A: address of the speech file

Unique ID:

Always a 4 digit number  
 English to Persian (Forward) Path:  $(n1 \text{ mod } 5000)$   
 Persian to English (Backward) Path:  $(n2 \text{ mod } 5000) + 5000$

Other messages: Starting and stopping the ASR can be accomplished by sending the messages through the GUI and including the 'start' and 'stop' commands as text (ex. FGATstart).

For the moment we only used the F: message flag and the T content flag.

Examples:

Say the N-best matches are (N=3)

My foot hurts p=0.5

I am sick p=0.2

I have the flu p=0.1

Conceptually:

FADT0012my\*0.5+foot\*0.5+hurts\*0.5|i\*0.2+am\*0.2+sick\*0.2|i\*0.1+have\*0.1+the\*0.1+flu\*0.1^

To avoid using characters that may conflict with future versions of USCPers+ we will exchange

Conceptual operation	Character code
*	\130
+	\131
	\132
^	\133

i.e.

**FADT0012my\130.5\131foot\130.5\131hurts\130.5\132i\130.2\131am\130.2\131sick\130.2\132i\130.1\131have\130.1\131the\130.1\131flu\130.1\133**

This will ensure that each block gets maximum information, and that each block has the freedom of internally processing that information in the best possible way.

Some comments:

1. In the absence of a DM module, the dummy module should just resend the same information with the new modified elvin code.
2. MT will be getting the N-best list but will only outputting a single phrase for the TTS system. If N-best is not useful for them then they can just ignore the extra N-1.