NLP challenge: JSGF development (Arum Park)

Task 1: Extend the English grammar

1.1 Extend the English JSGF grammar to cover at least the following utterances:

```
[i want to listen to]<unk> [jazz]<genre> [music]<unk>
[play me]<unk> [ummagumma]<album> [by]<unk> [pink floyd]<artist>
[put]<unk> [lady gaga]<artist> [on]<unk>
```

```
#JSGF V1.0 utf-8 en;

grammar music_play;

public <decl> = <declarative> <namedentity>;
public <int_imp> = <interrogative> <polite> <request> <namedentity>;
public <int_imp_phr> = <interrogative> <polite> <request_phrasal> <namedentity> (on);

public <declarative> = (I want to listen to);
public <interrogative> = [(can | could | would) you];
public <polite> = [please];
public <request> = (play [me | us]| put on);
public <request_phrasal> = (put);
public <namedentity> = <artist> | <song> | <genre> [music] | <album> [by <artist>];

<artist> =
the beatles |
radiohead |
lady gaga |
pink floyd;

<song> =
comfortably numb |
paranoid android |
let it be |
hey jude;

<genre> =
jazz |
hip hop |
rock;

<album>
ummagumma |
revolver|
ok computer|
the wall;
```

Task 2: Localize the JSGF grammar in your language (for Korean)

2.1 Localize the extended English grammar from Task 1.1 in your language (i.e., the language you are applying as a developer for). Feel free to add everything you think could help improve the quality of the generated utterances in your language.

```
#JSGF V1.0 EUC-KR;

grammar music_play;

public <decl> = <namedentity> <polite> <verbstem_orig> <periphrastic_want> <verbalsuffix_decl>;
public <imp> = <namedentity> <polite> <verbstem_variant> <verbalsuffix_imp>;
public <int> = <namedentity> <polite> <verbstem_variant> <periphrastic_can> <verbalsuffix_int>;

public <namedentity> = <artist> [의] [노래 | 앨범] | <song> | <genre> [음악] | [<artist> [의]] <album> [앨범];
public <polite> = [좀];

public <verbstem_variant> = (틀어 | 들려);   // put on (variant), play (variant)
public <verbstem_orig> = (듣);               // listen
public <periphrastic_can> = [줄 수 있)];     // can
public <periphrastic_want> = [고 싶)];               // want to
public <verbalsuffix_decl> = (어 [요] | 습니다);
public <verbalsuffix_int> = (어[요] | 니 | 을까[요]);
public <verbalsuffix_imp> = (줘 [요] | 주세요);

<artist> =
비틀즈 |          // the beatles
라디오헤드 |      // radiohead
레이디 가가 |     // lady gaga
핑크 플로이드;   // pink floyd

<song> =
comfortably numb | paranoid android | let it be | hey jude;

<genre> =
재즈 |  // jazz
힙합 |  // hip hop
락;     // rock

<album>
ummagumma | revolver| ok computer| the wall;
```

2.2 Provide some sample utterances the JSGF can produce using the localized grammar.

a.

비틀즈	노래	좀	듣고 싶어요
biteuljeu	nolae	jom	deud-go sip-eoyo
The Beatles	song	a bit	listen-PPC.WANT-SFS

'I want to listen to the song of the Beatles.'

b.

라디오헤드의	앨범	틀어줘
ladiohedeu-ui	aelbeom	teul-eojwo
Radiohead-POSS	album	put on-SFS.IMP

'Put on the album of Radiohead.'

c.

hey jude	좀	들려주세요
hey jude	jom	deullyeoju-seyo
Hey Jude	a bit	play-SFS.IMP

'Please play Hey Jude.'

d.

재즈	음악	좀	들려줄 수 있어
jaejeu	eumag	jom	deullyeoju-l su iss-eo
Jazz	music	a bit	play-PPC.POSSIBILITY-SFS

'Can you play some Jazz music?'

e.

ummagumma 앨범	틀어 줄 수 있을까
ummagumma aelbeom	teul-eo ju-l su iss-eulkka
Ummagumma album	put on-AUX-PPC.POSSIBILITY-SFS.INT

'Can you put on the Ummagumma album by Pink Floyd?'

2.3 What possible issues do you think you could run into if you were asked to extend the grammar considerably and how would you suggest overcoming them? There is no single correct response to this question so please share any potential issues that come to mind.

The original `<music_play>` rule could have been extended as below if we want to generate and cover only a definite set of English utterances:

```
public <music_play> =
[can you] [please] (put on | play [me]) (<artist>|<song>|<genre> [music] |<album> [by <artist>]);
```

However, it is not efficient to extend the grammar for other rules in this way because we always have to write every single rule and token whenever we want to reuse them for other rules. This is why I grouped the elements based on their linguistic properties and wrote the extended rules considering the effective expansion of the further rules for English. Since I used the linguistically universal features for the grouping, it could be easily used when the English grammar has to be extended to another language.

When I localized the English grammar in Korean, the Korean specific properties were taken into account: different word order in comparison to English (named entity first and the verbal phrase at the end), various verbal suffixes according to the sentence type and diverse combination of verb stem, auxiliary verb and verbal suffix to build a verbal complex.

2.4 Which features of your language complicate the localization or writing of the grammar? How would you solve or work around these complications?

a. In Korean, there are case particles for subject, object and topic. The grammar that I wrote for Korean does not generate any utterance that have those particles. The reason for that is that I wanted to keep the rules simple and it is because the case particles are optional especially for spoken Korean. But if we want to include case particles in the rules, their forms have to be taken into account because the forms differ according to their preceding words whether they end with consonant or vowel even for the same syntactic or pragmatic function:

syntactic/ pragmatic function	**environment**	**particle**
subject	consonant	이
	vowel	가
object	consonant	을
	vowel	를
topic	consonant	은
	vowel	는

e.g.

오랑우**탄** + **이**	강아**지** + **가**
(ㅌ + ㅏ + ㄴ)	(ㅈ + ㅣ)
ends with consonant	ends with vowel

This is why the different forms of the case particles need to be reflected in the rules when we want to generate utterances including those particles. What I can come up with for now is to write separate rules for consonant and vowel and to define the form of case particles according to the environment so that the utterances can be generated with the proper form of a case particle.

b. The word order of Korean language is quite free compared with English. This could be another challenge when writing context free grammar for Korean. Since the grammar only can generate utterances in the written order by the rules, I think it is hard to implement this property of Korean with the context free grammar especially when the utterance that we want to generate becomes longer in length.