The Samsung Speech Synthesis System for Blizzard Challenge 2023

Chunhui Lu\textsuperscript{1}, Jeasung Lee\textsuperscript{2}, Xue Wen\textsuperscript{1}, Xiaoyan Lou\textsuperscript{1}, Junkwang Oh\textsuperscript{2}

\textsuperscript{1}Samsung Research China – Beijing (SRC-B), China
\textsuperscript{2}Mobile eXperience Business, Samsung Electronics, Republic of Korea
{chunhui.lu, j-sung.lee, xue.wen, xiaoyan.lou, jun1.oh}@samsung.com

Abstract

This paper presents the Samsung text-to-speech system that participated in Blizzard Challenge 2023. This year’s challenge asks participants to build the voice from provided French data. It includes two tasks: a hub task with 50 hours audiobook data from a female native speaker and a spoke task with 2 hours speech data recorded by a second female speaker. Our system features a text analysis - acoustic model - vocoder pipeline. The text analyzer converts input text to a phoneme sequence including prosodic boundary and punctuation mark symbols. The acoustic model takes the phoneme sequence alongside contextual word embedding and sentence type to generate mel-spectrogram. It is built around FastSpeech, augmented with sentence-level and phone-level latent features to capture global and local prosodic variation. For vocoder, we use HiFi-GAN to reconstruct the waveform audio. We use the same solution for both tasks, each with its own training plan. Challenge results show that our system (identified as J) works well in similarity and intelligibility, which validates the effectiveness of our method.

Index Terms: Blizzard Challenge 2023, FastSpeech, HiFi-GAN

1. Introduction

The annual Blizzard Challenges are held to help better understand and compare research techniques in building corpus-based speech synthesizers on the same data. The challenges have witnessed the progress of text-to-speech (TTS) technology since 2005, from unit concatenation \cite{1, 2} to hidden Markov model (HMM) based statistical parametric speech synthesis (SPSS) \cite{3, 4} to latest neural end-to-end systems \cite{5, 6}. In the last challenge we had seen acoustic models based on FastSpeech\textsuperscript{7} and Tacotron2 \cite{8} dominate the challenge while GAN based neural vocoders such as MelGAN \cite{9} and HiFi-GAN \cite{10} surpass conventional methods like Griffin-Lim (GL) \cite{11}, STRAIGHT \cite{12} and WORLD \cite{13} in terms of speech quality and auto-regressive neural vocoders like WaveNet \cite{14} in efficiency.

This year’s challenge has two tasks:

- Hub task (2023-FH1: French TTS): Build a voice from 50 hours of provided audio book data from a female native speaker of French (NEB). Participants are allowed to use additional data provided it is publicly available.

- Spoke task (2023-FS1: Speaker adaptation): Build a voice that is closest to a second female native French speaker (AD) as possible. 2 hours of speech from AD is provided. There is no constraint on using additional data.

We participated in both tasks. Our system includes three parts: text analyzer, acoustic model, and vocoder. The text analyzer converts input text to a sequence of French phonemes. The acoustic model takes the phoneme sequence alongside contextual word embedding and sentence type as input to predict frame-level acoustic features, from which the vocoder reconstruct waveform audio. Our acoustic model is based on FastSpeech, augmented with sentence-level and phone-level latent features to capture global and local prosodic variations. For vocoder we use HiFi-GAN which offers well-balanced efficiency and quality. We use the same solution for both tasks, each with its own training plan.

In the following sections we describe our system and workflow in detail, and briefly summarize our results.

2. System description

We build our system in three stages: data preparation, model training, and synthesis. In data preparing stage, we extract mel-spectrograms and prepare annotations for phoneme sequences, phoneme durations, prosodic boundaries, liaisons and sentence types. In training stage we train several data driven models, including liaison prediction, break prediction, the main acoustic model, and the vocoder, using only officially provided data and labels derived from it. In synthesis stage we chain up all submodules into an automated pipeline that produced the final results for submission.

2.1. Data preparation

2.1.1. Mel spectrogram

Audio data provided by the challenge includes chapter-level audio files at 22.05 kHz sample rate and time-aligned textual transcriptions. We segment all audio files into one-sentence clips using time stamps from the transcription file. This results in a total of 63,942 utterances for task FH1 and 2,515 utterances for task FS1. We extract spectrograms using window size 1024, hop size 256, and DFT size 1024. These are then converted to mel-spectrograms with 80 frequency bands.

2.1.2. Phoneme sequence and duration

Hand-checked aligned phonetic transcriptions are available for a subset of NEB (44k utterances, about 30 hours) and for all AD utterances. These include the alignment of phonemes and corresponding durations for each character in the text. We use this information to align phonemes to characters throughout the text, excluding phonemes with duration 0. In addition, if a punctuation mark is aligned to the silence symbol ’\textdagger’ we replace the latter with the corresponding punctuation mark symbol. After generating the phoneme sequence, we convert the duration from milliseconds to the number of frames.
2.1.3. Prosodic boundary

TTS systems widely employ a hierarchical prosodic structure to distinguish different levels of breaks in a sentence. The prosodic boundary label tells how much break is expected at each word boundary. We annotate prosodic boundaries after each word at one of three levels: no break, short break and long break, according to the silence duration there, which is provided in the transcription file. The duration threshold to differentiate short break and long break is set at 170ms.

2.1.4. Liaison

In French phonology, liaison refers to the phenomenon of linking sounds between two words in specific contexts. It occurs when the current word ends with a consonant that is typically unpronounced, and the following word begins with a vowel. In such cases, the final consonant of the first word is pronounced, creating a smooth transition. Liaison contributes to the overall flow and rhythm of spoken French. We annotate these positions with the label ‘YES1’.

- Required liaisons: These are the positions where liaisons are mandatory by French grammar. We annotate these positions with the label ‘YES1’.
- Forbidden liaisons: These are the positions where liaisons should not be made under any circumstances. We annotate these positions with the label ‘NO1’.
- Optional liaisons: These are the positions where speakers have the choice of whether to pronounce a liaison or not. We annotate these positions with label ‘YES0’ or ‘NO0’, depending on whether the liaison is produced or not.

In total we annotate 2000 sentences for task FH1 and 800 sentences for task FS1.

2.1.5. Sentence type

The audio files of task FH1 are derived from audiobooks, which are rich in prosodic variations, particularly in the interlocution parts. We annotate each utterance with a binary sentence type flag to distinguish between dialogues and narrations. Since French orthography does not use quotation marks to delineate dialogues, we use ChatGPT to translate French text to English. French texts that align with English translation enclosed in quotation marks are then annotated as dialogues.

2.2. Model architectures and training

In this section we detail several predictive models we use for the challenge. These include break prediction, liaison prediction, the main acoustic model, and the vocoder. All models are initially trained on NEB data for task FH1, then fine-tuned on AD data for task FS1. Table 1 gives a summary of data usage in training these models, the remaining data is used for validation.

2.2.1. Break prediction

The break predictor predicts the prosodic boundary type (one of no break, short break, long break) of every word in the sentence. We use the Bidirectional Encoder Representations from Transformers (BERT) [15] model for this sequence tagging task. More concretely, we take an open source pre-trained French BERT model\(^1\) called CamemBERT [16] and fine-tune it on challenge scripts to predict the prepared prosodic boundary labels, supervised with cross-entropy (CE) objective. Training the break predictor took 4 epochs at learning rate of 5e-5.

2.2.2. Liaison prediction

The liaison predictor is responsible for predicting the liaison type, as defined in Section 2.1.4, for each marked word in the sentence. A word is marked if it ends with an unpronounced consonant and is followed by a word that starts with a vowel sound. Again we fine-tune the CamemBERT model on challenge scripts to predict prepared liaison labels. The model is trained with CE loss. Only marked words contribute to the loss calculation. In addition to the word and the binary marker, we also included a binary label that indicates the prosodic boundary type, i.e., whether there is a break after the word or not. This additional label is crucial because liaisons never occur at a prosodic break boundary. We trained the predictor for 5 epochs with a learning rate of 1e-4.

2.2.3. Acoustic model

Our main acoustic model follows our previous work [17, 18]. It is based on a variant of FastSpeech, as shown in Figure 1. The core part is an encoder-decoder DNN that converts a sequence of phonemes to a sequence of mel spectral frames. The length regulator matches their lengths by repeating encoder outputs. We replace the feed forward transformer blocks in both

\(^1\)https://huggingface.co/camembert-base

| Table 1: Data usage in training task specific models |
|---------------------------------|---------|---------|
| Model              | Task    | Data               |
| Break prediction   | FH1     | 40000 NEB utterances |
|                   | FS1     | 2250 AD utterances  |
| Liaison prediction | FH1     | 1800 NEB utterances |
|                   | FS1     | 700 AD utterances   |
| Acoustic model     | FH1     | 30 hours NEB data  |
|                   | FS1     | 2 hours AD data     |
| Vocoder            | FH1     | 50 hours NEB data  |
|                   | FS1     | 2 hours AD data     |

![Figure 1: Acoustic model structure. Dashed lines and modules are only used in training. Prosody predictor is trained separately.](https://example.com/image.png)
encoder and decoder with an improved conformer block in [19], and use relative positional encoding [20] to support paragraph-level synthesis.

We augment this basic model by introducing sentence-level and phoneme-level latent variables that conceptually capture unaccounted-for global and local prosodic variations. These latent features are extracted by two prosody extractors and join the main FastSpeech network at encoder output to condition the decoder. The sentence-level (global) prosody extractor follows the global style token (GST) [21] framework. This employs a reference encoder [22] to compress the spectrogram into a reference embedding, which then passes through a style token layer to generate the final style embedding. The phoneme-level (local) latent code is learned with the variational autoencoder (VAE) framework. Another reference encoder computes a variational posterior from phoneme-aligned spectrogram, from which a latent code was drawn. Both global and local latent codes are appended to the phoneme encoding before being sent to the length regulator.

We use adversarial training to address potential over-smoothing problem [23] and enhance the prediction of spectrograms. We use a multi-discriminator design following [24]. This divides the mel-spectrogram into three overlapping bands: low, medium, and high. Specifically, we assign the lowest 40 dimensions (from 0 to 40) as the low-frequency band, the middle 40 dimensions (from 20 to 60) as the mid-frequency band, and the highest 40 dimensions (from 40 to 80) as the high-frequency band. We use a separate discriminator for each of these bands. During training, each discriminator assesses a randomly selected fragment of the predicted or ground truth (GT) mel-spectrogram, where the length of the fragment could vary.

The objective function of acoustic model (generator) is the combination of reconstruction and adversarial losses:

$$\mathcal{L}_G = \mathcal{L}_{rec} + \mathcal{L}_{adv_G}$$

The reconstruction loss $\mathcal{L}_{rec}$ takes the form of an evidence lower bound (ELBO) under standard Gaussian latent prior:

$$\mathcal{L}_{rec} = \mathbb{E}_{q(x|z)}[\log p(x|y, z, s)] - \lambda_{KL} \sum_{u=1}^{U} D_{KL}(q(z_u|x)||\mathcal{N}(0, I))$$

where $x$, $y$, $z$, $s$ represent the GT mel-spectrogram, the sequence of input phonemes, the sequence of local latents and the global style token, respectively. $z_u$ is the latent code for the $u$-th phoneme. $U$ is the number of phones. We use $0<\lambda_{KL}<1$, which favors accuracy over latent space exploration.

We used the LS-GAN [25] loss as the adversarial objective:

$$\mathcal{L}_{adv_G} = \frac{1}{3} \sum_{f \in F} \mathbb{E}_{\hat{x}}[(D_f(\hat{x}) - 1)^2]$$

$$\mathcal{L}_{adv_D} = \sum_{f \in F} [\mathbb{E}_{x}(D_f(x) - 1)^2 + \mathbb{E}_{\hat{x}} D_f(\hat{x})^2]$$

where $F = \{\text{low, mid, high}\}$ is the set of frequency bands, $\hat{x}$ is the predicted mel-spectrogram, $D_f$ is the discriminator for frequency band $f$.

After training the enhanced FastSpeech model, we collect global and local latent codes for each utterance and phoneme, then train a separate prosody predictor that predicts these latent codes as well as phoneme durations from text. This prosody predictor provides prosodic features during synthesis. Input features of the prosody predictor include the phoneme sequence, contextual word embeddings from the last layer of CamemBERT, and the sentence type label in section 2.1.5. For the global style token we pool word embeddings into a sentence embedding, which is concatenated to the sentence type label to form a sentence-level input. For phoneme-level durations and latent codes we repeat each word embedding by the number of phonemes it contains to form phone-level inputs. The overall objective function of the prosody predictor is:

$$\mathcal{L}_p = \mathcal{L}_{sen} + \mathcal{L}_{pha} + \mathcal{L}_{dur}$$

where $\mathcal{L}_{sen}$ is the L1 loss between predicted and extracted global style tokens, $\mathcal{L}_{pha}$ is the KL (Kullback-Leibler) divergence each phoneme’s predictive distribution from its extracted variational posterior, $\mathcal{L}_{dur}$ is the mean square error (MSE) between predicted and annotated phoneme durations on the logarithmic scale.

The FastSpeech encoder, decoder, and the phoneme encoder in prosody predictor each use 4 conformer block layers with attention dimension 256 and a hidden convolutional feed-forward module of size 1024. The prosody extractors extract a 128-dim global style embedding and a 3-dim latent code for each phoneme. The phonemes follow the structure of variance adapter in [26]. We set $\lambda_{KL}$ to 0.01. All discriminators share the same model structure but differ in model parameters, each with three 2D-convolution layers followed by a leaky ReLU activation function and a linear projection for final output. Training the TTS engine, including both the enhanced FastSpeech model and the prosody predictor, took about 300k steps at batch size 48 on one NVIDIA A6000 GPU.

2.2.4. Vocoder

We use HiFi-GAN v1 version for reconstructing speech waveform from mel-spectrogram. The model is trained using the official implementation2. It is initially trained from scratch on NEB data for task FH1 and then fine-tuned on AD data for task FS1.

2.3. Synthesis

In the synthesis stage we chain up individual modules into an automated pipeline for text-to-speech generation. In summary, the text analyzer converts input text to a phoneme sequence including prosodic boundary and punctuation mark symbols, following the format described in Section 2.1.2; the acoustic model takes these alongside word embeddings and the sentence type label (section 2.1.5) to generate mel-spectrogram; the vocoder converts the mel-spectrogram to waveform audio. The model size and running time of each module trained in Section 2.2 are listed in Table 2. The runtime was calculated by executing each model on 129 SUS test sentences using one A6000 GPU with a batch size of 1.

**Table 2: Model size and running time of each module trained in Section 2.2**

<table>
<thead>
<tr>
<th>Model</th>
<th>Size</th>
<th>Runtime</th>
</tr>
</thead>
<tbody>
<tr>
<td>Break Predictor</td>
<td>449 MB</td>
<td>1.16 s</td>
</tr>
<tr>
<td>Liaison Predictor</td>
<td>451 MB</td>
<td>1.21 s</td>
</tr>
<tr>
<td>Acoustic Model</td>
<td>184 MB</td>
<td>4.15 s</td>
</tr>
<tr>
<td>Vocoder</td>
<td>53 MB</td>
<td>1.63 s</td>
</tr>
</tbody>
</table>

---

2https://github.com/jik876/hifi-gan
2.3.1. Text analysis

Text analysis include break prediction, grapheme-to-phoneme (G2P) conversion, homograph disambiguation, and liaison prediction. The workflow is shown in Figure 2. G2P and homograph disambiguation modules are shared by the two tasks.

The procedure for generating the final pronunciation of each word is as follows.

1. We use an internal French G2P module to generate phoneme sequences, which are then mapped to officially provided phoneme symbols. This was a letter-to-sound model based on the classification and regression tree (CART) [27]. For each letter of a word the model predicts the corresponding phoneme deterministically according to the letter itself and three contextual letters on either side (7 in total).
2. We apply homograph disambiguation to words with more than one possible pronunciations, then replace the G2P result above with the disambiguated result. See below 2.3.2.
3. We apply liaison prediction to marked words. If a word is predicted with liaison label ‘YES1’ or ‘YES0’, we add the sound of the ending consonant, listed in Table 3, to the phoneme sequence obtained from the previous two steps.

Table 3: Ending consonant letter and its corresponding sound

<table>
<thead>
<tr>
<th>Letter</th>
<th>Sound</th>
<th>Letter</th>
<th>Sound</th>
</tr>
</thead>
<tbody>
<tr>
<td>d</td>
<td>t</td>
<td>s</td>
<td>z</td>
</tr>
<tr>
<td>t</td>
<td>t</td>
<td>x</td>
<td>z</td>
</tr>
<tr>
<td>c</td>
<td>k</td>
<td>z</td>
<td></td>
</tr>
<tr>
<td>q</td>
<td>k</td>
<td>y</td>
<td></td>
</tr>
<tr>
<td>g</td>
<td>k</td>
<td>p</td>
<td></td>
</tr>
<tr>
<td>n</td>
<td>n</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.3.2. Heterophonic homograph disambiguation

A heterophonic homograph refers to one of two or more words that are spelled alike but have different meanings or pronunciations. An example in French is the word “fils,” which can mean son in “mon fils” or threads in “des fils”. There are 803 heterophonic homographs in French. We use a homograph lexicon to identify these words, and implement several disambiguation strategies to differentiate between candidate pronunciations, as follows.

For the majority of homographs, especially those ending with “-tions,” “-ent,” or “-er,” the pronunciation can be determined by their part of speech (POS). We include word, POS, and pronunciation entries in the lexicon for these words. We use a pretrained French POS tagging model to predict the POS of each homograph word, then retrieve from the lexicon the pronunciation matching the predicted POS.

For homographs that cannot be disambiguated solely based on the POS, we expand the lexicon to include their meanings in English. At synthesis time we use ChatGPT to translate the homograph word into English then retrieve the pronunciation matching the English meaning.

Among the 36 homograph words in final test set, 4 words were disambiguated by ChatGPT while the remaining words were disambiguated by POS tagging. Several homograph words are given in their lexicon form in Table 4.

Table 4: Word examples in constructed homograph lexicon

<table>
<thead>
<tr>
<th>Word</th>
<th>POS</th>
<th>Lemma</th>
<th>Meaning</th>
<th>Pronunciation</th>
</tr>
</thead>
<tbody>
<tr>
<td>actions</td>
<td>n.</td>
<td>-</td>
<td>-</td>
<td>a k s j o -</td>
</tr>
<tr>
<td>content</td>
<td>v.</td>
<td>-</td>
<td>k o - t a -</td>
<td></td>
</tr>
<tr>
<td>reporter</td>
<td>n.</td>
<td>-</td>
<td>r x p o ' r t e r</td>
<td></td>
</tr>
<tr>
<td>convent</td>
<td>v.</td>
<td>-</td>
<td>k o - v i</td>
<td></td>
</tr>
<tr>
<td>fils</td>
<td>n.</td>
<td>-</td>
<td>k o - v j e -</td>
<td></td>
</tr>
</tbody>
</table>

3. Evaluation results

18 teams have submitted results for task FH1 and 14 have submitted for task FS1. Speech examples submitted by the participants, baselines (identified as BF and BT), and natural speech (identified with letter A) are evaluated on several metrics. Our system is identified with letter J.

3.1. Quality and similarity

Mean opinion score (MOS) for quality and voice similarity are evaluated for both tasks. For quality evaluation, listeners are asked to rate the quality of the synthetic voice on a scale from 1 [Very poor] to 5 [Excellent]. For similarity, listeners are asked to compare the synthetic voice against 4 reference samples, then rate how similar they sound on a scale from 1 [Sounds like completely a different person] to 5 [Sounds like exactly the same person]. Figure 3 and Figure 4 show the scatter plots of quality and similarity scores for task FH1 and task FS1, respectively.

In task FH1, our system achieves a quality MOS of 3.7 and a similarity score of 3.3. In task FS1, the quality and similarity scores are 3.9 and 4.2, respectively. Our system ranks in the middle among all systems in terms of quality, and this can be attributed to two possible reasons. Firstly, the quality degrades due to the mismatch between training and inference in separately trained acoustic models and vocoders. More importantly, our limited understanding of French prevents us from effectively evaluating the synthesized speech, identifying potential issues and making necessary optimizations during system development. Regarding similarity, although our system did not achieve the highest score, there are no significant differences between our system and those with higher MOS scores.

3.2. Homograph disambiguation

Homographs are words that are spelled alike but have different meanings. In French, there are 803 such homographs. We use a homograph lexicon to identify these words, and implement several disambiguation strategies to differentiate between candidate pronunciations, as follows.

For the majority of homographs, especially those ending with “-tions,” “-ent,” or “-er,” the pronunciation can be determined by their part of speech (POS). We include word, POS, and pronunciation entries in the lexicon for these words. We use a pretrained French POS tagging model to predict the POS of each homograph word, then retrieve from the lexicon the pronunciation matching the predicted POS.

For homographs that cannot be disambiguated solely based on the POS, we expand the lexicon to include their meanings in English. At synthesis time we use ChatGPT to translate the homograph word into English then retrieve the pronunciation matching the English meaning.

Among the 36 homograph words in final test set, 4 words were disambiguated by ChatGPT while the remaining words were disambiguated by POS tagging. Several homograph words are given in their lexicon form in Table 4.

Table 4: Word examples in constructed homograph lexicon

<table>
<thead>
<tr>
<th>Word</th>
<th>POS</th>
<th>Lemma</th>
<th>Meaning</th>
<th>Pronunciation</th>
</tr>
</thead>
<tbody>
<tr>
<td>actions</td>
<td>n.</td>
<td>-</td>
<td>-</td>
<td>a k s j o -</td>
</tr>
<tr>
<td>content</td>
<td>v.</td>
<td>-</td>
<td>k o - t a -</td>
<td></td>
</tr>
<tr>
<td>reporter</td>
<td>n.</td>
<td>-</td>
<td>r x p o ' r t e r</td>
<td></td>
</tr>
<tr>
<td>conven</td>
<td>v.</td>
<td>-</td>
<td>k o - v i</td>
<td></td>
</tr>
<tr>
<td>fils</td>
<td>n.</td>
<td>-</td>
<td>k o - v j e -</td>
<td></td>
</tr>
</tbody>
</table>

3.3. Quality and similarity

Mean opinion score (MOS) for quality and voice similarity are evaluated for both tasks. For quality evaluation, listeners are asked to rate the quality of the synthetic voice on a scale from 1 [Very poor] to 5 [Excellent]. For similarity, listeners are asked to compare the synthetic voice against 4 reference samples, then rate how similar they sound on a scale from 1 [Sounds like completely a different person] to 5 [Sounds like exactly the same person]. Figure 3 and Figure 4 show the scatter plots of quality and similarity scores for task FH1 and task FS1, respectively.

In task FH1, our system achieves a quality MOS of 3.7 and a similarity score of 3.3. In task FS1, the quality and similarity scores are 3.9 and 4.2, respectively. Our system ranks in the middle among all systems in terms of quality, and this can be attributed to two possible reasons. Firstly, the quality degrades due to the mismatch between training and inference in separately trained acoustic models and vocoders. More importantly, our limited understanding of French prevents us from effectively evaluating the synthesized speech, identifying potential issues and making necessary optimizations during system development. Regarding similarity, although our system did not achieve the highest score, there are no significant differences between our system and those with higher MOS scores.
We have observed that the similarity score in task FH1 is noticeably lower than that in task FS1, even when considering natural speech. This indicates that the speaker NEB's timbre is more diverse, possibly due to role-playing in audiobooks or her personal choices in articulation variances.

3.2. Intelligibility

Two Intelligibility evaluations are conducted for task FH1. For intelligibility test of semantically-unpredictable sentences (SUS), listeners are asked to listen to each sentence only once and type in what they heard. Among all systems, our system achieves the lowest word error rate at 9.5%.

In the intelligibility evaluation of sentences including homographs, listeners are presented with three audio samples: one synthesized sample and two reference samples, which include different pronunciation versions of the homograph. They are then asked to choose the reference audio that best matches the pronunciation of the homograph in the synthesized sample. Our system achieved the accuracy of 94% among the 144 test samples. By analyzing the per homograph pronunciation accuracy, we found that the errors arise from either inaccurate predicted POS or incorrect pronunciation entries in homograph lexicon, which are sourced from Wiktionary.

4. Conclusion

In this paper, we have presented our TTS system developed for Blizzard Challenge 2023. The system is built following text analysis - acoustic model - vocoder pipeline. The text analyzer converts input text to a sequence of French phonemes including prosodic boundary and punctuation mark symbols. The acoustic model takes this phoneme sequence alongside word embedding and sentence type as input to predict mel-spectrogram. It is built based on FastSpeech with sentence-level and phoneme-level prosody modelling to capture global and local prosodic variations, followed by a HiFi-GAN vocoder. We employ a unified solution for both tasks, with each task having its own train-

5. References


