POSCAT: A Morpheme-based Speech Corpus Annotation Tool

Byeongchang Kim, Jin-seok Lee, Jeongwon Cha, Geunbae Lee

Department of Computer Science & Engineering
Pohang University of Science & Technology
Pohang, 790-784, South Korea
{bckim, wolfpack, jwcha, gblee}@nlp.postech.ac.kr

Abstract

As more and more speech systems require linguistic knowledge to accommodate various levels of applications, corpora that are tagged with linguistic annotations as well as signal-level annotations are highly recommended for the development of today's speech systems. Among the linguistic annotations, POS (part-of-speech) tag annotations are indispensable in speech corpora for most modern spoken language applications of morphologically complex agglutinative languages such as Korean.

Considering the above demands, we have developed a single unified speech corpus annotation tool that enables corpus builders to link linguistic annotations to signal-level annotations using a morphological analyzer and a POS tagger as basic morpheme-based linguistic engines. Our tool integrates a syntactic analyzer, phrase break detector, grapheme-to-phoneme converter and automatic phonetic aligner together. Each engine automatically annotates its own linguistic and signal knowledge, and interacts with the corpus developers to revise and correct the annotations on demand. All the linguistic/phonetic engines were developed and merged with an interactive visualization tool in a client-server network communication model.

The corpora that can be constructed using our annotation tool are multi-purpose and applicable to both speech recognition and text-to-speech (TTS) systems. Finally, since the linguistic and signal processing engines and user interactive visualization tool are implemented within a client-server model, the system loads can be reasonably distributed over several machines.

1. Introduction

As statistical methods have become dominant in speech research communities, large annotated speech corpora have become essential for various speech systems. In case of speech recognition systems, large speech corpora tend to promise good performance, regardless of whether the corpus is phonetically aligned or not. The parameters of Hidden Markov Model (HMM) and bigram, trigram or n-gram language models can be well estimated with large corpora. In case of TTS systems, a speech corpus with linguistic annotations is required to predict prosodic elements like intonation, pause and duration. Furthermore, a thoroughly phonetically aligned speech database, which can be extracted from naturally or carefully spoken speech, makes TTS systems more intelligible.

To build a large speech corpus with its linguistic and signal annotations, corpus builders use annotation tools that can reduce cumbersome and time-consuming tasks. The annotation tools must help the builders creat large and linguistically annotated corpora rapidly and accurately using a set of functions, such as signal processing, linguistic processing, grapheme-to-phoneme conversion, automatic phonetic alignment, and even language model generation, each of which is unique to each tool. However, most previous speech annotation tools only deal with signal and phonetic level tagging, and have been developed for a single type of application domain. As speech systems increasingly require linguistic knowledge to accommodate various levels of applications, corpora that are tagged with linguistic annotations, as well as signal-level annotations, are highly recommended for development of today's speech systems. Accordingly, we propose a speech corpus annotation tool that enables corpus builders to link linguistic annotations to signal-level annotations in corpora using several linguistic and signal processing engines. Each engine automatically annotates its own linguistic and signal knowledge and interacts with corpus builders to accommodate revisions and corrections of the annotations on demand. A corpus constructed using this annotation tool is multi-purpose and applicable to both speech recognition/understanding and text-to-speech (TTS) systems.

Among the linguistic annotations, POS (part-of-speech) tag annotations are indispensable in speech corpora because such morphemic annotations are essential for most modern spoken language applications of morphologically complex agglutinative languages such as Korean. In the case of grapheme-to-phoneme conversion, because phonological changes may occur in phonologically conditioned environments as well as in morphologically conditioned environments, phonological and morphological knowledge should be merged and used together in a grapheme-to-phoneme converter. The statistical language model, which can be used in automatic speech recognition systems, should be constructed both at the morpheme-level and at the word-level with POS tags, graphemes and phonemes in order to accommodate the agglutinative characteristics of Korean.

In the next section, existing speech corpus tools are reviewed and compared with our system. Section 3 describes the design philosophy of our speech annotation tool (POSCAT: POSTECH Corpus Annotation Tool) and Section 4 briefly explains several signal and linguistic processing engines. A client visualizing tool is described in Section 5, and some conclusions are drawn in Section 6.

2. Previous Research

Most previous annotation tools have been developed for a single type of application domain. Table 1. shows existing speech corpus annotation tools and their characteristics. The first two columns show the names of systems or projects and their developers, and the last three

System/ Project	Developer	Natural Language Processing	Phonetic Segmentation	Segmentation Units
Annotator	Entropic	Dictionary based grapheme-to-phoneme conversion	Automatic with the Aligner	Phone
Archivage	LACITO/CNRS	No	Manual	Sentence (sentence level)
CHILDES	CMU	Morphosyntactic analyzer	Automatic (word level)	Word, sentence and discourse
SoundWalker/ CSAE	University of California Santa Barbara	No	No	Sentence
CSLU Toolkit	OGI	No	Manual	Phone
Segmenter	ISIP	No	Manual (word level)	Word
SFS	University College London	No	Manual	Phone
SLAM	Institute of Phonetics and Dialectology	No	Automatic	Phone
Snack	Sjolander	No	Manual	Phone
Speech Analyzer	SIL	No	Automatic	Phone
Transcriber	DGA	No	Manual	Phone, word and sentence
Praat	Paul Boersma	No	Manual	Phone, word and sentence
POSCAT (our system)	**POSTECH**	**POS tagging, grapheme-to-phoneme conversion, syntactic Analysis**	**Automatic**	**Phone, word and sentence**

Table 1: Existing speech annotation tools and their characteristics

columns show their characteristics comparing with our POSCAT system. All of them visualize signal waves and show their textual transcriptions and some other information as necessary. Most of them perform signal processing to help corpus builders annotate signal and linguistic knowledge on the speech corpus, where only two systems utilize natural language processing like grapheme-to-phoneme conversion and morphosyntactic analysis (Entropic, 1997; CMU, 1998). Though most of them support phonetic/word-level/sentence-level segmentation, only the Annotator, the SLAM and the Speech Analyzer automatically segment speech waves into phonetic units (Entropic, 1997; Institute of Phonetics and Dialectology, 1997; SIL, 1999). CHILDES supports word-level automatic segmentation, Archivage and Segmenter support sentence-level and word-level manual segmentation, and the others support only manual phonetic segmentation (CMU, 1998; Michailovsky et al., 1998; ISIP, 1999).

In summary, there are no tools that have the ability to either manually or automatically annotate morphemes with their corresponding POS tags and phoneme sequences, which are indispensable in speech corpora for agglutinative languages, such as Korean, Japanese, Finnish, Turkish, etc.

3. POSCAT Design Philosophy

There are many kinds of speech corpora in the speech research community. The major usages of the corpora are to support development of speech recognition systems, to provide prosodic elements for TTS systems, to give phonetic segments to a speech signal synthesizer in TTS systems, and to provide a variety of speaker or language modeling for speaker or language recognition systems, and so on. According to their usages, each corpus has its own characteristics, such as recording environments, number of speakers, narrative/fluent, overlapping/nonoverlapping speech fragments, and so on.

The purpose of the corpora that can be constructed by our speech corpus annotation tool is to support the development of both automatic speech recognition/understanding and TTS systems. For the development of conventional automatic speech recognition systems, a large and phonetically aligned speech corpus is necessary in training a HMM, and a large POS tagged and error-free text corpus is required to generate a statistical language model. The development of conventional TTS systems requires two speech corpora. One is small and composed of phonetically well-aligned speech segments, and the other is large and prosodically annotated with POS tags and parse trees.

We can construct the corpus for an automatic speech recognition system using the following steps. First, textual transcriptions and their speech signals are prepared. Second, POS tagging and grapheme-to-phoneme conversion are performed on the textual transcriptions, sentence by sentence, because grapheme-to-phoneme conversion requires the results of POS tagging. Third, we can now complete the corpus by aligning the phonetic labels with their corresponding speech segments.

The conventional sequence of making speech corpora for TTS systems is as follows. The small and phonetically well-aligned speech corpus is constructed by transcription preparing, speech recording and phonetic aligning, without any linguistic processing. The large prosodically annotated corpus can be manually constructed by syntactic analyzing, phrase break detection and manual prosodic labeling on the corpus that was constructed for an automatic speech recognition system.

As described in the previous paragraphs, most of the tasks required to build annotated speech corpora are tedious and time-consuming. Our annotation tool can accelerate this process by helping corpus builders annotate signal and linguistic knowledge on the speech corpus easily, precisely and rapidly. The following are the design parameters of our speech corpus annotation tool.

- The speech corpus annotation tool has to browse the corpus and visualize some portion of the corpus in various ways as demanded by tool users.
- The speech corpus annotation tool has to annotate the signal and linguistic knowledge on the corpus automatically although the annotations are not so precise.
- The automatically assigned linguistic annotations must include POS tags that provide a basic level of syntactic classes for each morpheme in morphologically complex agglutinative languages.
- The speech corpus annotation tool has to provide corpus builders with a facility to revise and correct the automatically annotated corpus.
- The speech corpus annotation tool must not overload a machine in order to provide the corpus builders with various signal and linguistic knowledge.

We designed a speech corpus annotation tool to accommodate the above design parameters. The single unified speech corpus tool enables corpus builders to link linguistic annotations to signal-level annotations using a morphological analyzer and a POS tagger as basic morpheme based linguistic engines, and integrates a syntactic analyzer, phrase break detector, grapheme-to-phoneme converter and automatic phonetic aligner together. First, each engine automatically annotates its own linguistic and signal knowledge. Second, the visualization tool interacts with corpus developers to revise and correct the annotations on demand. All

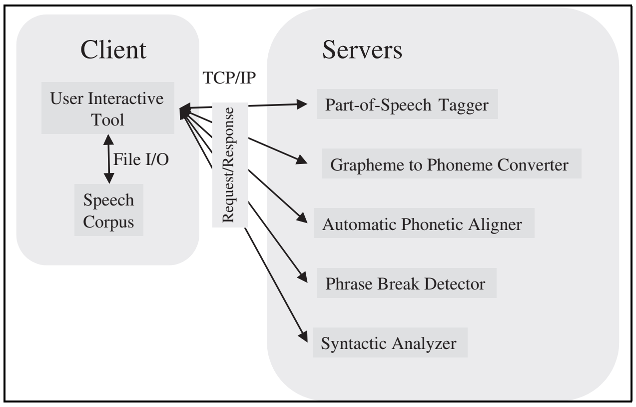

Figure 1: Linguistic and signal processing engines and visualization tool using a client-server communication model

the linguistic/phonetic engines were developed and merged with an interactive visualization tool using a client-server communication model to distribute system loads over several machines.

4. Linguistic and Signal Annotation Servers

There are 5 server engines for linguistic and signal processing in POSCAT, including morphological analyzer and POS tagger, syntactic analyzer, phrase break detector, grapheme-to-phoneme converter and automatic phonetic aligner. As shown in the previous section, each engine plays an important role in signal and linguistic annotating. The engines are distributed over several machines and merged with an interactive visualization tool using a client-server communication model as shown in Figure 1. The protocol in which the servers and the client communicate is the simplest: the client requests and then the servers respond through the TCP/IP layer without any error control. The following give a brief explanation about each server engine.

Morphological Analyzer and Part-of-Speech Tagger There are three major components in our hybrid architecture for Korean POS tagging with generalized unknown-morpheme guessing: the morphological analyzer with unknown-morpheme handler, the statistical POS tagger, and the rule-based error corrector (Cha et al., 1998).

Grapheme-to-Phoneme Converter Our grapheme-to-phoneme conversion method uses a dictionary-based and rule-based hybrid method with a phonetic pattern dictionary and CCV (consonant consonant vowel) LTS (letter to sound) rules (Kim et al., 1998).

Automatic Phonetic Aligner The aligner uses a phone-based Hidden Markov Model (HMM) and Viterbi search algorithm without any complex entities. The aligner dynamically strings together the phonetic HMMs in the sequence determined by the phonetic transcription, and finds the optimal time alignment between the phonetic transcription and the waveform using the Viterbi search algorithm.

Phrase Break Detector Our current phrase break detector consists of a probabilistic phrase break detector and a transformational rule-based post error corrector (Kim and Lee, 1999). The probabilistic phrase break detector segments the POS sequences into several phrases according to word trigram probabilities. The initial phrase break tagged morpheme sequence is corrected with post error correcting rules.

Syntactic Analyzer The approach we have developed combines the advantages of CCG's ability of type raising and compositions along with abilitiy of variable categories and unordered arguments modeling for relatively free word order treatment (Lee et al., 1994; Lee et al., 1997). In KCCG, type-raising using case-markers is adopted for converting nouns into the functors over a verb, and a composition rule is used for coordination modeling (Cha et al., 1999).

5. Client Visualization Tool

The client visualization tool reads data from files, constructs internal data structures from them and displays them. It also consults all the linguistic servers located in different machines concerning the linguistic annotations of the given speech and text, and serves corpus builders to annotate, revise and correct signal and linguistic annotations easily and consistently.

We developed the client visualization tool with the scripting language Tcl/Tk and C extensions (Figure 2). It utilizes the Snack sound extension, which has primitives for sound visualization (Sjolander, 1999).

We now describe some required functions of the client visualization tool, file format in which the annotations are stored physically, and data structures in which the annotations are stored logically.

5.1. Functions

The visualization tool has to provide all the data in visual form on demand and help users to annotate some markers. The types of data to be displayed are wave, textual transcription, spectrogram, zero crossing rate, power, POS tag sequence, phonetic alignment with corresponding phonetic transcription, parse tree and phrase break sequence.

A wave and its textual transcription are basic data from external sources. Simple signal analyses such as the FFT, power computation and Zero Crossing Rate (ZCR) are performed by the client visualization tool because the analyses require all the waves and produce results of the same size as the waves or bigger. The size of waves and their results are much heavy to be communicated via network. The other linguistic data are delivered by the linguistic server engines and are revised by the corpus builders, so the client tool contains only functions with which the corpus builders trigger the server engines to produce and revise the linguistic data.

5.2. File Formats for Annotations

There have been as many file formats for annotated speech corpora as there have been speech tools. Though each has its own strong points, the overhead costs to support these formats are not so small. We decided to use XML markup as the file format for our speech corpus annotation tool, which made it possible to use existing knowledge and software, and thus maximize the portability. There are also many file formats using XML markup, and UTF is a representative one (NIST, 1998). However, because UTF is not appropriate for accommodating linguistic data, such as POS tags, phonetic time-aligns and syntactic categories, some tags for linguistic annotations and their structures are newly defined. Figure 3 is an example of our annotation file.

```
<corpus name="MBC NEWSDESK">
  <section name="980919">
    <sentence name="98091920">
      <text>지금 중학교 3학년 학생부터 적용될 새 대입제도 개선안이 나왔습니다.</text>
      <speech>
                <Xref xml:link="simple" href="./pcm/98091920.pcm"/>
      </speech>
      <phrase index="1"> long
        <word index="1">
          <morpheme index="1">
            <POStag>MCC</POStag>
            <root>지금</root>
            <surface>지금</surface>
            <phoneme index="1">ㅈ
              <time start="110" end="175"/>
            </phoneme>
            <phoneme index="2">ㅣ
              <time start="175" end="200"/>
            </phoneme>
            <phoneme index="3">ㄱ
              <time start="200" end="220"/>
            </phoneme>
            <phoneme index="4">ㅡ
              <time start="220" end="260"/>
            </phoneme>
            <phoneme index="5">ㅁ
              <time start="260" end="325"/>
            </phoneme>
          </morpheme>
        </word>
        <word index="2">
          <morpheme index="1">중학교
            <POStag>MCC</POStag>
            <root>중학교</root>
            <surface>중학교</surface>
            <phoneme index="1">ㅈ
              <time start="325" end="370"/>
            </phoneme>
              ......
          </morpheme>
        </word>
        ......
      </phrase>
      <phrase index="2">
        ......
      </phrase>
        ......
      <parse_tree>
                <Xref xml:link="simple" href="./parse_tree/980919.xml#98091920"/>
      </parse_tree>
    </sentence>
    <sentence name="98091920">
      ......
    </sentence>
  </section>
  <section name="980919">
    ......
  </section>
</corpus>
```

Figure 3: An example of our annotation file

5.3. Internal Data Structures for Annotations

Our fundamental structure of a corpus is a tree. The corpus consists of several sections, each section consists of several sentences, and a sentence consists of several phrases, which in turn consist of several words comprising one or more morphemes. Though it is possible to represent them in a graph structure as in (Bird and Liberman, 1999), we adopted tree structures as the fundamental internal annotation structures, and added list structures to link the entities in the same layer.

Figure 4 shows the overall data structures used in our client tool. The corpus node is a root node of the entire structure, where all the entities are structured hierarchically and all the entities in the same layer are linked sequentially. Because a parse tree is irrelevant to the phrases located between the sentence node and word nodes, the tree is located independently with the other annotation structure. The sentence node has a link to the root node of the parse tree corresponding to the sentence, and the leaf nodes of

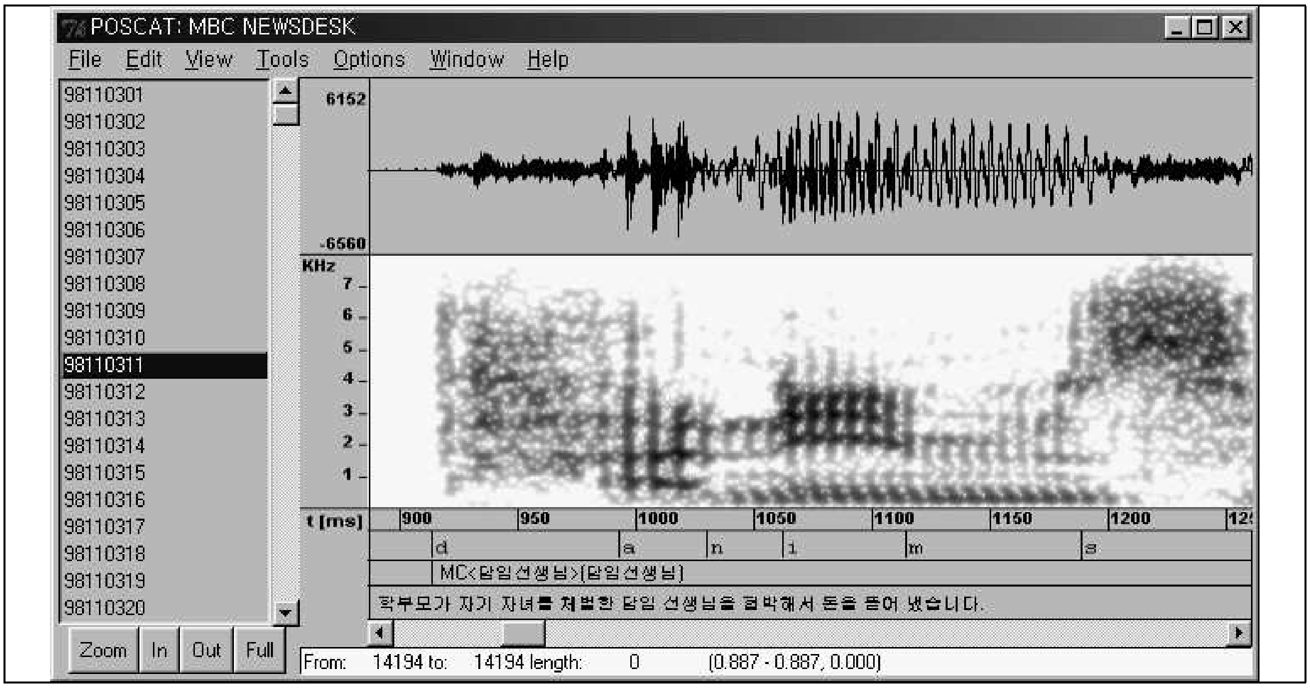

Figure 2: The client visualization tool

the parse tree have links to the corresponding morphemes. Each node, except nodes in the parse tree, has its own time indexes.

There is a conventional problem when using tree structures to store annotations. Insertion or deletion of some layers requires reconstruction of the tree structures to maintain consistency. In the case of our annotation structure, this is not the case because all the layers constituting the tree structures are prepared automatically by the server engines and subsequently no layer deletion exists.

6. Conclusion

We have developed a unified speech corpus annotation tool integrated with a morphological analyzer and a POS tagger, syntactic analyzer, phrase break detector, grapheme-to-phoneme converter and automatic phonetic aligner. Therefore, the annotation tool can automatically annotate not only signal-level annotations but also linguistic annotations, and corpus builders can link linguistic information to signal-level information, and can revise and correct the annotations.

Moreover, the annotation tool facilitates POS (part-of-speech) and syntactic tag annotations that are indispensable in speech corpora, because they provide basic levels of syntactic classes for each morpheme. Such morphemic annotations are essential for most modern spoken language applications of morphologically complex agglutinative languages.

The corpora that can be constructed using our annotation tool are multi-purpose and applicable to both speech recognition and TTS systems. The phonetically aligned and POS tagged speech corpus is essential in all speech recognition systems, while phrase breaks and morphologically/syntactically aligned speech corpora are very useful in prosody and pronunciation generation for every TTS system.

Finally, since the linguistic and signal processing engines and user interactive visualization tool are implemented using a client-server model, the system loads can be reasonably distributed over several different machines.

7. Acknowledgements

This paper was supported by the University Research Program of the Ministry of Information & Communication through the IITA(1998.7-2000.6).

8. References

Bird, Steven and Mark Liberman, 1999. A formal framework for linguistic annotations. Technical Report MS-CIS-99-01, Department of Computer and Information Science, University of Pennsylvania.

Cha, Jeongwon, Geunbae Lee, and Jong-Hyeok Lee, 1998. Generalized unknown morpheme guessing for hybrid POS tagging of Korean. In *Proceedings of the Sixth Workshop on Very Large Corpora*.

Cha, Jeongwon, WonIl Lee, Geunbae Lee, and Jong-Hyeok Lee, 1999. Morpho-syntactic modeling of Korean with K-CCG. In *Proceedings of the International Conference on Computer Processing of Oriental Language*.

CMU, 1998. *CHILDES*. http://childes.psy.cmu.edu.

Entropic, 1997. *Annotator*. http://www.entropic.com/products&services/annotator/annotator.html.

Institute of Phonetics and Dialectology, 1997. *SLAM*. http://nts.csrf.pd.cnr.it/IFeD/Pages/slam.htm.

ISIP, 1999. *Segmenter tool*. http://www.isip.msstate.edu/projects/speech/software/swb_segmenter/index.html.

Kim, Byeongchang and Geunbae Lee, 1999. Statistical/rule-based hybrid phrase break detection.

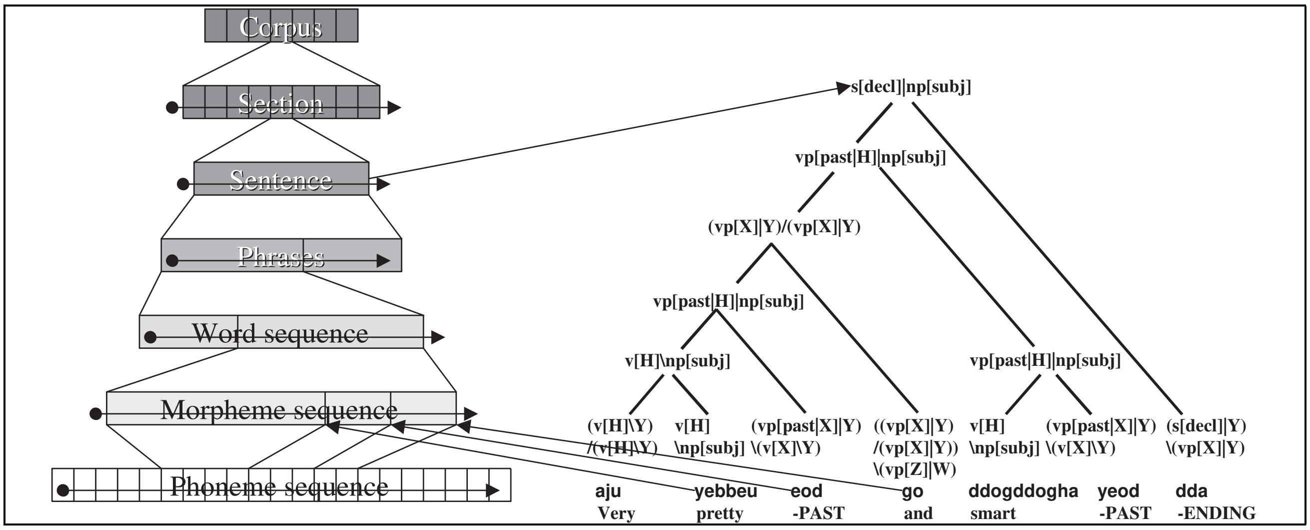

Figure 4: Internal data structure for corpus annotations

In *Proceedings of the International Conference on Speech Processing*.

Kim, Byeongchang, WonIl Lee, Geunbae Lee, and Jong-Hyeok Lee, 1998. Unlimited vocabulary grapheme to phoneme conversion for Korean TTS. In *Proceedings of the Coling-ACL '98*.

Lee, WonIl, Geunbae Lee, and Jong-Hyeok Lee, 1994. Table-driven neural syntactic analysis of spoken Korean. In *Proceedings of COLING-94*.

Lee, WonIl, Geunbae Lee, and Jong-Hyeok Lee, 1997. Morpho-syntactic modeling of korean with a categorial grammar. In *Proceedings of the natural language processing pacific-rim symposium*. Phuket, Thailand.

Michailovsky, Boyd, John B. Lowe, and Michel Jacobson, 1998. *Archivage*. http://lacito.vjf.cnrs.fr/ARCHIVAG/ENGLISH.htm.

NIST, 1998. A universal transcription format (UTF) annotation specification for evaluation of spoken language technology corpora. Technical Report www.nist.gov/speech/hub4_98/utf-1.0-v2.ps, NIST.

SIL, 1999. *Speech Analyzer*. http://www.sil.org/computing/sil_computing.html.

Sjolander, Kare, 1999. *The Snack Sound Extension for Tcl/Tk*. http://www.speech.kth.se/snack.