Karlsruhe Institute of Technology

Sub-word Language Models for German LVCSR

Diploma Thesis of

Adnene Zairi

At the Department of Informatics
Interactive Systems Lab (ISL)
Institute for Anthropomatics and Robotics

Reviewer:	Prof. Dr. Alexander Waibel
Second reviewer:	Dr. Sebastian Stüker
Advisor:	Dipl.-Inform. Kevin Kilgour

Duration: 01 December 2014 – 31 May 2015

I declare that I have developed and written the enclosed thesis completely by myself, and have not used sources or means without declaration in the text.

Karlsruhe, 31.05.2015

. . . . . .

(**Adnene Zairi**)

Abstract

This work is devoted to build sub-word language models for German Large Vocabulary Continuous Speech Recognition (LVCSR) Systems. The motivation of using a sub-words based system comes from its ability to model unseen compound words in the training corpus by compounding sequences of sub-units forming new words. The two techniques we apply for the generation of sub-words, are based on syllables and letter n-grams. In three scenarios, we investigate different techniques acting on the input split vocabulary and the vocabulary for the language model and the pronunciation dictionary. Using a combination of full-words and syllables from the training corpus presents the most efficient split method, which also leads to the best word error rate (WER) results during the decoding task. This split method consists on keeping the most frequent words in the training corpus and splitting the infrequent ones into syllables or characters.

Zusammenfassung

Diese Arbeit beschäftigt sich mit dem Bauen automatischen Spracherkennungssystemen für die Deutsche Sprache basierend auf Sub-word Sprachmodelle. Da die Deutsche Sprache eine morphologisch reiche Sprache ist, ist es unmöglich für die Trainingsdaten alle möglichen Wörter zu beinhalten. Aus diesem Grund entscheiden wir uns für die Benutzung von Sub-words um die ungesehene Wörter im Trainingsset zu erkennen. Silben und Buchstaben n-Gramme sind zwei Techniken für die Erzeugung von Sub-words. In drei Szenarien untersuchen wir verschiedene Techniken zum Einsatz von Sub-words in der Spracherkennung, wo die Wörter-Splitmethoden sich ändern. Hierbei werden unterschiedliche Vokabulare für Sprachmodelle und Aussprachewörterbücher verwendet. Eine Kombination aus Voll-Wörter und Silben aus dem Trainingskorpus führt zum besten Splitten des Trainingssets und entsprechend zu der besten Wort Fehler Rate (WER) beim Dekodieren. Die verwendete Split Methode besteht darin, die häufigsten Wörter zu behalten und alle restlichen Wörter in Silben oder Buchstaben aufzuteilen.

Contents

1. Introduction

The technological development leap in the past 30 years was marked by using a complex and innovative machines in research. Input devices such as keyboard or remote control are needed to manipulate these machines, but human target to communicate without input through keyboard. Today it is possible to use speech which is the most natural way of human communication to communicate with machines.
Speech technologies allow people around the world to participate in the information revolution and to link people together, helping to overcome language barriers.

Automatic Speech Recognition (ASR) is one of these technologies which allow a smooth and comfortable human-machine interaction. It can be used in different domain such as in-car systems, telephony, education and daily life. But ASR is available only for a few languages comparing to about 7106 known living languages in the world [LSF14].

The ASR system considered in this thesis is a large vocabulary continuous speech recognition (LVCSR) system. LVCSR system is able to deal with a large vocabulary of words more than 100k pronounced continuously in a fluent manner. An preprocessing is needed in LVCSR system to convert the speech signal into a sequence of feature vectors. Thereafter looking for the highest probable sequence of words that leads to the acoustic features by using a statistical approach.

1.1 Motivation

The German language is known by rising types of lexical as a large number of distinct lexical forms. Word compounding, inflection, and derivation are the most factors that generated this phenomena. These 3 types of morphological processes can be defined as follows [Fea12]:

- Derivation: add affixes to form new words.
- Inflection: the formation of grammatical variants of a word.
- Word compounding: join (compose) words together to form new words.

Hence, the rich morphology nature of German language is one of the main difficulties concerning the German Large Vocabulary Continuous Speech Recognition (LVCSR) systems based on whole-words or full-words as vocabulary units. The fundamental problems of these full-words speech recognition systems are that the training data and Speech data can not contain all words in vocabulary.

To avoid these problems, LVCSR systems can use the sub-word language models based on word decomposition or word splitting into fragments. Sub-word units occur more frequently and can be trained more robustly than words. They also offer the possibility to deal with the challenge of unseen words in training data by compounding sequences of sub-units to form new words. Consequently the improvement of the word error rate (WER) which is the standard evaluation measure for LVCSR systems. Basically, the goal of the ASR system is to minimize WER measured on the decoded output.

1.2 ASR Challenges

An ASR system aims to transcribe an unknown spoken utterance into its most likely written word sequence. The major challenges in unrestricted, continuous speech recognition are [Mou14]:

- No indication of the word or sub-word boundaries in the acoustic signal.
- There is a considerable variation in the speaking rates in continuous speech.
- Words are pronounced inaccurately in fluent speech and the most of these are the word endings.
- Environmental noise affect on the quality of the speech signal.
- The recognition system should take into account the correlation between semantic and syntactic shape of the language in the same manner to human-to-human communication.

1.3 Structure of the Thesis

After the introduction in Chapter 1 and an overview of the ASR system and his different components. We introduce the tools that we have used for our experiments. Then an overview of the related work given in Chapter 3 with a definition of some sub-word units. A brief overview of the linguistic characteristics of German language are presented in the chapter 4. In Chapter 5 , we give a detailed description of our experimental setup. In Chapter 6, we discuss our experiments and evaluate the results. Finally, the Chapter 7 summarizes our work and present future perspectives.

2. Fundamentals

This chapter contains basic information helpful for reading this thesis. In Section 2.1, an overview of a typical speech recognition system is given and the different components are defined. In 2.1.3.2, evaluation criteria for system and LM performance are described. In Section 2.2, the used tools are introduced.

2.1 Automatic Speech Recognition

Automatic speech recognition (ASR) has the role to convert human speech signal into written text with the help of an automatic process through computer. The human speech is recorded with a microphone and used as input speech signal of the traditional ASR process. The basic architecture of an ASR system is presented in Figure 2.1. The speaker

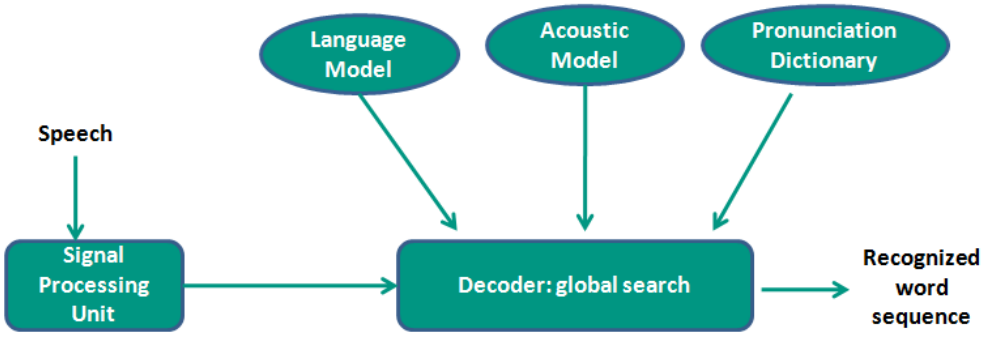

Figure 2.1: Architecture of ASR system.

uses his or her apparatus to produce speech, which as signal through the signal processing unit captured. The parameterization of observed acoustic signal into a sequence of acoustic vectors $\mathbf{X} = \mathbf{X_1 X_2 ... X_n}$ is made by an acoustic analysis. Our goal is to define the most likely word sequence $\mathbf{W} = \mathbf{W_1 W_2 .. W_m}$ in relation to our acoustic observation X

and the acoustic model and language model. Moreover, with the ASR process we have to find the word sequences that maximize this $\mathbf{P}(\mathbf{W}|\mathbf{X})$.

To achieve this, we apply Bayes rule 2.1 [BAY58] to rewrite the word sequence in the following form 2.2, which recapitulates the computational model used for large vocabulary continuous speech recognition (LVCSR) and decomposes the required probability $\mathbf{P}(\mathbf{W}|\mathbf{X})$ into two components.

$$\mathbf{P}(\mathbf{W}|\mathbf{X}) = \frac{\mathbf{P}(\mathbf{X}|\mathbf{W})\mathbf{P}(\mathbf{W})}{\mathbf{P}(\mathbf{X})} \tag{2.1}$$

$$\begin{aligned} \hat{\mathbf{W}} &= \underset{\mathbf{w}}{\mathbf{argmax}}\ \mathbf{P}(\mathbf{W}|\mathbf{X}) \\ &= \underset{\mathbf{w}}{\mathbf{argmax}}\ \frac{\mathbf{P}(\mathbf{X}|\mathbf{W})\mathbf{P}(\mathbf{W})}{\mathbf{P}(\mathbf{X})} \\ &= \underset{\mathbf{w}}{\mathbf{argmax}}\ \mathbf{P}(\mathbf{X}|\mathbf{W})\mathbf{P}(\mathbf{W}) \end{aligned} \tag{2.2}$$

In Equation 2.1 the probability $\mathbf{P}(\mathbf{X}|\mathbf{W})$ represent the acoustic models which is the representation of the knowledge about acoustics, phonetics, gender and dialect differences among speakers. Since there is a large number of words, we try to decompose them into a sub word units when creating an acoustic model. This procedure is very closely related with the phonetic modeling.

Generally, the language model captures the linguistic properties of the language and provides the A-priori-Probability $\mathbf{P}(\mathbf{W})$ of a word sequence $\mathbf{W}$. The language model works one level higher and observes the relations between words. Since some words are more likely to co occur with others and also the occurrence sequence plays a role, all this additional information is present in the language model.

The decoder, which tries to find $\hat{\mathbf{W}}$ in 2.2, uses both acoustic and language models and searches the most probable word sequence as a result. Because of the dictionary or lexicon, this defines the mapping from words to sub-word units, usually phonemes. is the search limited.

The ultimate goal of the speech research in the last years is to have an apparatus to understand fluently spoken speech. Despite ASR technology is well developed in many fields and applications, machines still find difficulty to transcribe speech if we change the acoustic environment or the speaker.

Therefore, the objective of current ASR research is the recognizing of speech through machines with 100% accuracy. Even words are spoken by speakers different in age, sex and with different accent. Also, regardless of vocabulary size and the existing unwanted sound in the environment. Today, accuracy can reach 90% or more if you trained only an individual speaker's voice and if there is a lot of vocabularies.

Acoustic model, pronunciation dictionary and language model, which are the basis three components of an ASR system, are described in more detail in the following sections.

2.1.1 Acoustic Model

An acoustic model (AM) is used in ASR to determine statistical representations of each of the audio signal (distinct sounds) that represent a word W in term of feature vectors. A phoneme calling each set of these statistical representations which represent a smallest unit of Language and can change the meaning of the word. The German language has about 25 distinct sounds that are useful for speech recognition, and thus we have 25 different phonemes. But due to co-articulation there are many more sounds that occur when speaking German.

As observed in Section 2.1, the objective of the acoustic modeling is to compute the likelihood that the observation of a sequence of an acoustic vectors $\mathbf{X} = \mathbf{X_1 X_2 ... X_n}$ will be produced for a given word sequence $\mathbf{W} = \mathbf{W_1 W_2 .. W_m}$. Therefore, we need to have a large speech corpus and the statistical representations for each phoneme in a language. These statistical representations are created by using a special training algorithms and are called Hidden Markov Model (HMM), which is the most popularly used phoneme model in the LVCSR system and each phoneme has its own HMM.
HMM,

The Figure 2.2 shows a three-state-left-to-right HMM typically used in speech recognition.
A Hidden Markov Model $\lambda = (A, B, \Pi)$ can be defined as a five-tuple consisting of [RJ86]:

- **S**: is the set of States $S = \{s_1, s_2, ..., s_n\}$ with n is the number of states.
- Π: is the initial probability distribution, $\Pi(s_i) = P(q_1 = s_i)$ probability of s_i being the first state of a sequence
- **A**: is the matrix of state transition probabilities: 1 ≤ i, j ≤ n
  $A = (a_{ij})$ with $a_{ij} = P(q_{t+1} = s_j | q_t = s_i)$ going from state s_i to s_j.
- **B**: is the set of emission probability distributions/densities, $B = \{b_1, b_2, ..., b_n\}$ where $b_n(x) = P(o_t = x | q_t = s_i)$ is the probability of observing x when the system is in state s_i.
- **V**: is the alphabet of possible emitted feature vectors.
  The observable feature space can be discrete $V = \{x_1, x_2, ..., x_v\}$, or continuous $V = R^d$.

According to [Rab89], Evaluation, Decoding, and Learning problems are the fundamental problems of HMM.

Evaluation Problem which is to calculate the probability of the model that λ has generated sequence X by giving the HMM $\lambda = (A, B, \Pi)$ and the observation sequence X. This problem is solvable with Forward Backward Algorithm.

Decoding problem aims to give the HMM $\lambda = (A, B, \Pi)$ and the observation sequence X to calculate the most likely sequence of hidden states that produces this observation. Viterbi Algorithm can solve this problem.

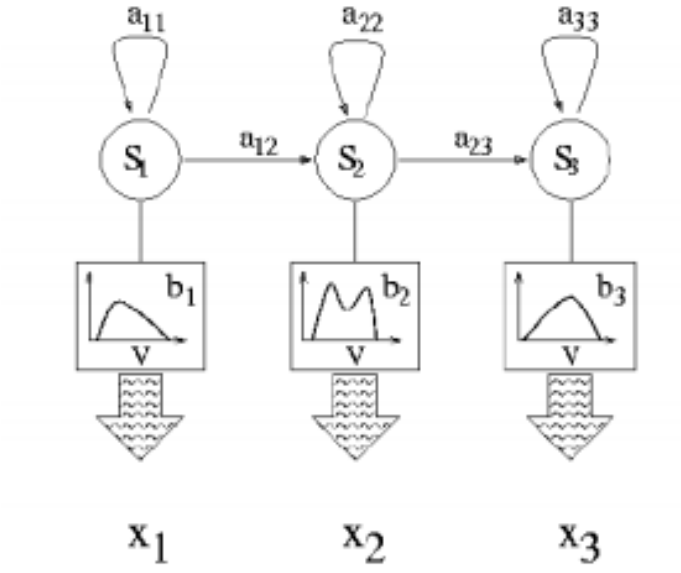

Figure 2.2: HMM, generating an observation of feature vectors $X = X_1X_2X_3$
[SK06]

Learning problem which is optimizing the parameters of λ so that the probability of observing the vector sequence X is maximized by giving the HMM $\lambda = (A, B, \Pi)$ and the observation sequence X. BaumWelch algorithm is a solution for this problem and it is considered the traditional method for training HMM.

We can use different typologies such as: Linear model, Bakis model, left-to-right model, Alternative paths, Ergodic model to design a HMM. In ASR we apply the Bakis model [Bak76] for each section of a phone (begin -b, middle -m, end -e) to represent each HMM-state model. The Figure 2.3 shows how to connect the HMMs together to form words

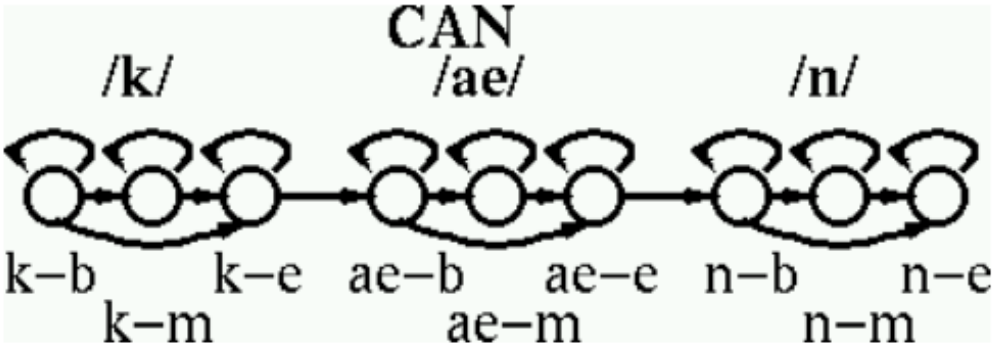

Figure 2.3: HMM for the word "can"
[Sch08]

for example "can" . Similarly, words can be joined together to cover complete utterances [You96]. However, depending on the context where it occurs or when spoken by a different speaker the same phoneme sounds different. For example, the phoneme /s/

sounds different in the two words "sound" and "this". Therefore, to obtain a good phonetic discrimination we need to train different HMMs for each different context. Here we talk about Allophone models if a specific phoneme has too much possible HMMs. Benjamin Lee Whorf was the first to use the term allophone in 1940 which is a set of multiple possible spoken sounds or phones used to pronounce a single phoneme in a particular language [Tra59] [Lee96].

Moreover, the simplest and most common strategy is to use triphones models as a group of three phone, where each phone has a distinct HMM for each unique pairs of left and right neighbors [You96]. A triphone represent a three linear HMMs states and the possible transitions are the loop, the forward and the skip transition [SN02].

Due to the large variation of triphones, a lot of them may not been observed enough often in the training data. This can be solved by the using of phonetic decision trees which is an automatic method for modeling the context dependence of pronunciation, to estimate which phones sounds similar in different context based of questions about the left and right neighbors [BdSG+91] [YOW94]. Here we talk about context-dependent phoneme model which is usually used by the modern LVCSR systems.

Furthermore, linguistic knowledge are needed to choose the context questions. According to [GO15] these questions may include tests for:

- A specific phone, phonetic classes such as stop and vowel.
- More restrictive classes for example voiced stop and front vowel.
- More general classes like voiced consonant.

Typically, there are about 100 questions for each context (left vs. right).

Figure 2.4 displays how to use a decision to cluster the center state of some /e/ triphones.

Clustering algorithm is needed for the phonetic discrimination as follows [Sch12]:

1. Initialize one cluster containing all contexts (or join all context in one cluster)
2. For all clusters: compute distance of subclusters
3. Perform the split that get the largest distance (information gain)
4. Continue with step 2 until satisfied (number of clusters)

HMMs can be divided into two distinct categories: continuous HMM and discrete HMM, while the emission probabilities themselves can be discrete. In ASR the discrete HMMs are infrequently used [MR+02].

The Gaussian Mixture Models (GMMs) represents the emission probability, which is probability density functions, for continuous HMMs. The problem is that this approach requires a larger set of training data since there are many parameters to be estimated.

Depending on the tying degree of the Gaussians in the system we distinguish between Fully-continuous HMMs, where each model has its own codebook of Gaussians and Semi-continuous HMMs, where one codebook of Gaussians are shared by all models.
As shown in Figure 2.5 for the Fully-continuous HMMs, the states that correspond to the

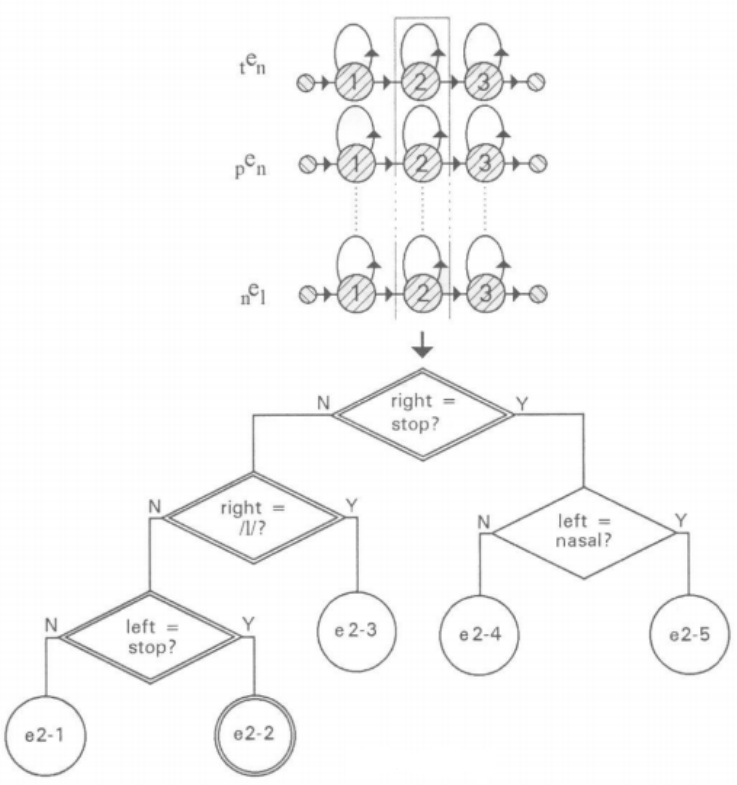

Figure 2.4: Decision tree used to cluster the center state of some /e/ triphones [Hol01]

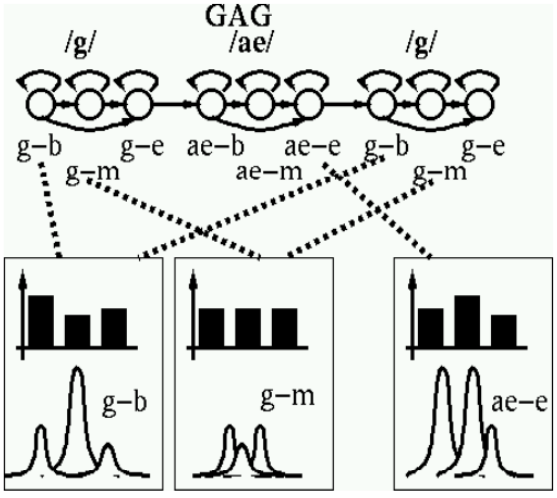

Figure 2.5: Example for Fully-continuous HMMs [Sch08]

same acoustic phenomenon share the same acoustic model. The parameters of the emission probabilities can be estimated more robustly and the training data can be exploited better. But it requires a larger set of training data since there are many parameters to be estimated [Sch08].

The semi-continuous HMMs aims to solve this problem by using parameter tying to share

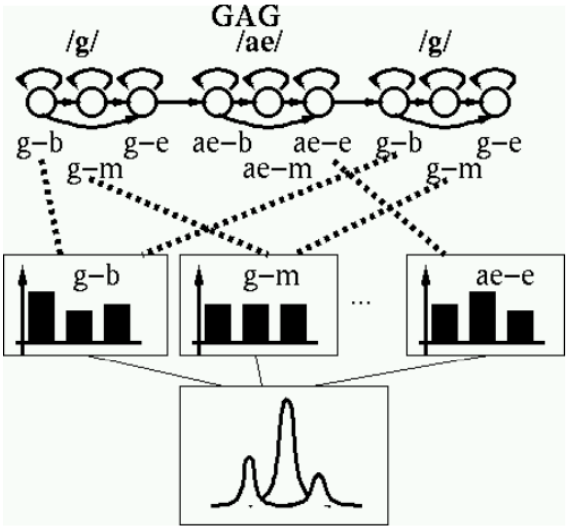

Figure 2.6: Example for semi-continuous HMMs
[Sch08]

more data between the parameters as illustrated in Figure 2.6, where there is only one codebook of Gaussians in the system. Every acoustic model has its own set of mixture weights, but shares the same Gaussian codebook [Sch12]. This approach reduces the amount of parameters that is estimated enormously and offers a compromise between accuracy and trainability.

2.1.2 Pronunciation Dictionary

The pronunciation dictionary or lexicon contains a list of words with associated pronunciation represented as a combination of phonemes.

Generally most words have a single pronunciation. Multiple pronunciations, pronunciation variants, are allowed to account for pronunciation variability [HHSL05]. However in speech recognition, many pronunciation variants cause recognition errors in the form of deletions, insertions or substitutions of phoneme [MGSN98].

To represent the phoneme we need a standard phonetic alphabet such as SAMPA (Speech Assessment Methods Phonetic Alphabet) [W$^+$97] or IPA (International Phonetic Alphabet) [Ass99]. SAMPA is a machine-readable phonetic alphabet based on ASCII-7 symbol set (codes) and IPA is a an alphabetic system of notation for sounds of languages based on the Latin alphabet.

As illustrated in Figure 2.7 the generation of the pronunciations dictionary can be statistical or rule-based. Also rule-based can be completely manual or manually supervised. By completely manual we need experts in linguistics to type the phone sequence for each given word. If the generation is manually supervised, we create rules from an existing dictionary to product pronunciations of new entries. In this method to provide pronunciations for inflected forms and compound words, a reasonably sized starting dictionary is required [Sch12].

Moreover the statistical method is often based on the sequitur Grapheme-to-Phoneme (G2P) method [BN08]. The basic principal is to apply graphone (or grapheme-phoneme

joint-multigram) approach to the alignment problem and to use standard language modeling techniques to model transcription probabilities [BN03]. The most challenge of the

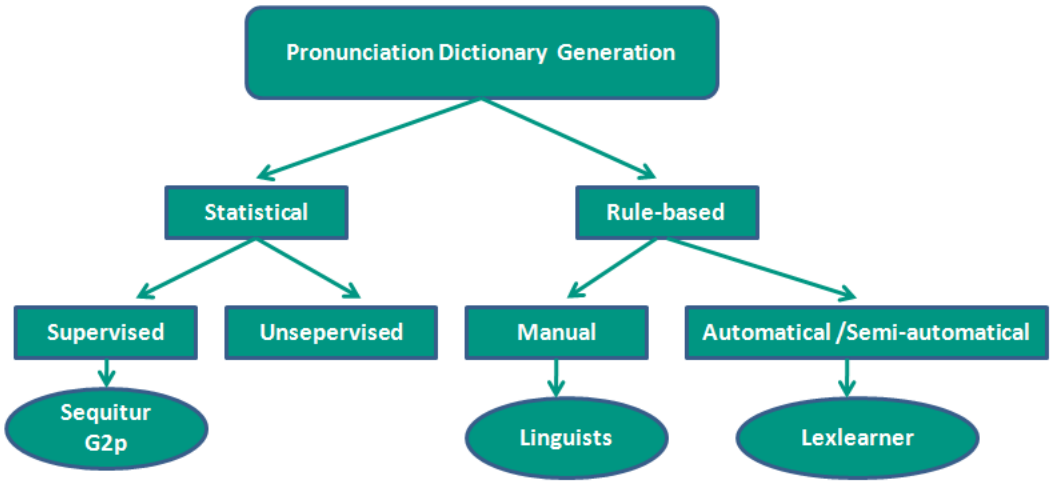

Figure 2.7: Pronunciation Dictionary Generation.
[Sch12]

production of the pronunciation modeling is the pronunciation variants. It exits more than one pronunciations for the same word and it depends on the context or co-articulation effects, dialects, accents and emotions [Sch12]. This problem can be solved if we have the possibility to add and to mark multiple pronunciation variants of the word in the dictionary as seen in this example of one German word:

durch (1) {{ D WB } U ER { CH WB }}
durch (2) {{ D WB } U I { CH WB }}
durch (3) {{ D WB } U R { CH WB }}
durch (4) {{D WB } U R { X WB }}

In this example the markers of the beginning and end of words WB (Word Boundary) are clearly specified for the entry of more than one phoneme.

The size of the dictionary varies from few to millions of words and depends on the application and the language. Thousand words and above usually used for speech recognizers for LVCSR.

2.1.3 Language Model

The language model provides the A-priori-Probability of a word sequence $\mathbf{W} = \mathbf{w_1}, \mathbf{w_2}, ..., \mathbf{w_n}$. This corresponds to $\mathbf{P(W)}$ in Equation 2.1. Using the definition of conditional probability, the probability of a word sequence can be written as follows:

$$\mathbf{P(W) = P(w_1, w_2, ..., w_n) = \prod_{i=1}^{n} P(w_i | w_1, ..., w_{i-1})} \tag{2.3}$$

The goal of the language model in speech recognizer application is to calculate the probabilities $\mathbf{P(w_i | w_1, ..., w_{i-1})}$ for all possible next word in a context called history H (for a word to be seen in a defined context).

For example, if in the English language the context is "I live in a white" then the probability of the next word being "house" is a lot higher than the probability of the next word being "mouse" [Kil09].

Even for a large vocabulary size V there is a huge number of possible histories, when computing **P(w| H)**. Since many of the word sequences in the history will never be observed or would be observed very few times, it is not possible accurately to estimate this probabilities [Par11].

To come over this, choose for vocabulary the most frequent set of words in the training text and group the histories into a tractable number of equivalence classes $\mathbf{P(w|H) = P(w|\Phi(H))}$. Equation 2.3 can then be rewritten as follows:

$$\mathbf{P(W) = \prod_{i=1}^{n} P(w_i | w_1, ..., w_{i-1}) = \prod_{i=1}^{n} P(w_i | \Phi(w_1, ..., w_{i-1}))} \tag{2.4}$$

where $\Phi : h \rightarrow C$ associates a history h to an equivalence class belonging to a finite set C [BBV04].

Two typical types of language models are the n-gram LM, which uses a probabilistic approach, and the Context Free Grammar LM, which is based on formal languages. N-gram LM is presented in the following paragraph.

2.1.3.1 Statistical Language Models (N-grams)

The probability of a word in N-gram language models depends only on the n-1 previous words. Therefore the equivalence class in equation 2.4 can be simply based on the several previous words. We talk about a trigram if the word depends on the previous two words $\mathbf{P(w_i | w_{i-1}, w_{i-2})}$. Similarly, we can have unigram: $\mathbf{P(w_i)}$, or bigram: $\mathbf{P(w_i | w_{i-1})}$ language models. The trigram is particularly powerful, as most words have a strong dependence on the previous two words, and it is estimated reasonably well with an attainable corpus [ID10]. Generally a language models contain special symbols indicating the start **< s >** and end **< /s >** of a sentence. **< /s >** is necessary at the end of the sentence to make the sum of the probabilities of all strings equal 1 [ID10]. As a word sequence "< s > I live in a white" may have a relatively high probability but its probability as a sentence "< s > I live in a white < /s >" will be very low. For example, the probability of sentence "I live in a white house" is given in the equation 2.5. The sentence is split into the trigrams.

- <s> <s> I
- <s> I live
- I live in
- live in a

- in a white
- a white house
- white house </s>
- house </s> </s>

$$\mathbf{P(I, live, in, a, white, house) = P(I| < s >, < s >) * P(live| < s >, I) * P(in|I, live)} \\ \mathbf{*P(a|live, in) * P(white|in, a) * P(house|a, white)} \tag{2.5}$$

In equation 2.6 a trigram LM is used to calculate the probability of a word, which depends on its two preceding words that can be seen in a defined context.

$$\mathbf{P(w_i|w_{i-2}, w_{i-1}) = \frac{\#(w_{i-2}, w_{i-1}, w_i)}{\#(w_{i-1}, w_i)}} \tag{2.6}$$

where #(X) is the number of times the specified sequence of words (X) occurs in the training data.

Generally a large training set of text that contains million words is necessary to train a n-gram model. In fact the number of possible n-grams can be increased exponentially with relation to n. For example for V vocabulary of word and for n=3, we have a V^3 possible trigrams. For typical LVCSR there is V^3 a very high number of trigrams. Sometimes many possible n-grams are not seen or in some cases few appears in the training data. The effect of unseen n-gram in a word sequence of the test data is to have a probability of zero. That leads to sparsity problem, which is the hard problem in n-gram modeling. Therefore various smoothing techniques have been developed to guarantee that all possible word combinations are assigned nonzero probabilities [SDSV] [Mou14]. In section 2.1.3.3 we define smoothing techniques with more details.

2.1.3.2 Evaluation

The standard evaluation metric of an automatic speech recognition is the word error rate (WER) [Dou98], which is based on the Levenshtein distance [Lev66], also called the edit distance. WER is the percentage of word errors in the hypothesis sentence compared to the reference sentence. The following errors can occur after the alignment of the hypothesis and the reference text [Sch14]:

- Substitution: A wrong word is recognized.
- Deletion: A word from the reference is missing in the hypothesis.
- Insertion: The recognizer inserts a word that is not actually spoken.

Figure 2.8 shows the possible errors output of an automatic speech recognition system. The vertical axis represents the reference, and the horizontal output sequence of the system [Sch14].

Search for a given reference the minimum number of insertion i, deletion d and substitution s needed to transform the hypothesis into the reference. The equation 2.7 of WER after identifying the errors is estimated as follows:

$$\mathbf{WER = \frac{\#substitutions + \#deletions + \#insertions}{\#words(reference)} * 100} \tag{2.7}$$

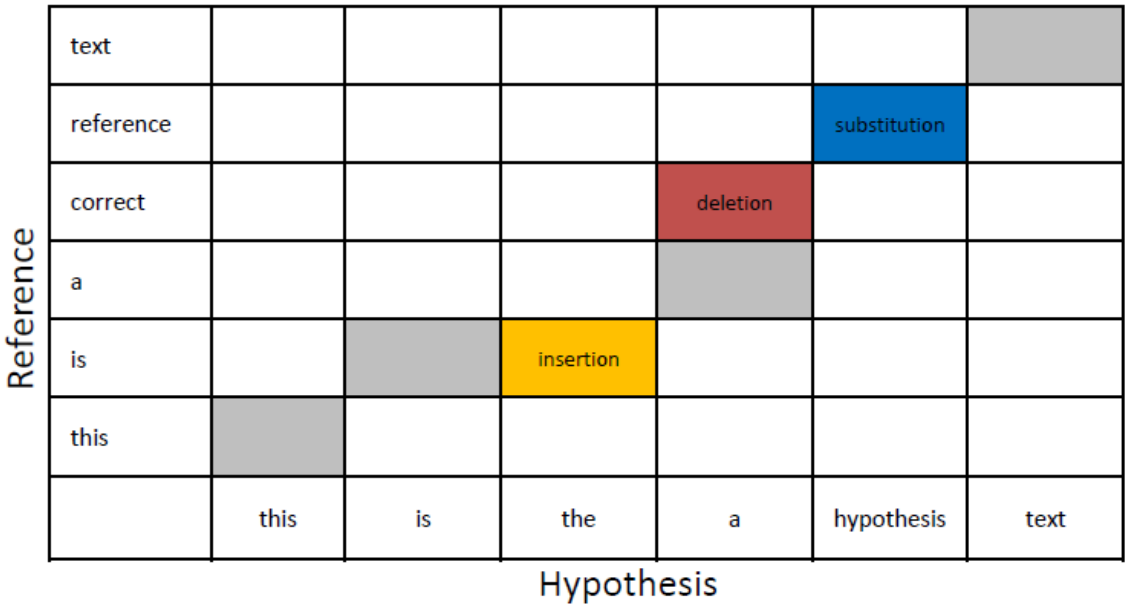

Figure 2.8: Possible errors of an automatic speech recognition system. [Sch14]

To evaluate the performance of a language model, a metric called perplexity is used. The perplexity is defined as $\mathbf{2^{H_p(T)}}$, where $\mathbf{H_p(T)}$ is the cross-entropy of the language model on a set of test sentences T, containing $\mathbf{|T|_w}$ words.

$$\mathbf{H_p(T)} = \frac{\sum_{\mathbf{t \in T}} \log_{\mathbf{2}} \mathbf{P(t)}}{\mathbf{|T|_w}} \tag{2.8}$$

$$\mathbf{ppl} = \mathbf{2^{H_p(T)}} \tag{2.9}$$

A low perplexity is an indicator that the language model is good. [KP02] shows that lower perplexity values correlate with lower word error rates.
The combination of two different language models, called linear interpolation, can improve the performance of speech recognition. The probability $\mathbf{P(w_a, h_a)}$ of a word w_a given a history h_a for the interpolated languages models L1 and L2 is represented as follows:

$$\mathbf{P(w_a, h_a)} = \mathbf{(1-\lambda) P_{L1}(w_a, h_a) + \lambda P_{L2}(w_a, h_a), 0 \leqslant \lambda \leqslant 1} \tag{2.10}$$

where λ is the interpolation weights, which is automatically calculated to decrease the perplexity of a set of a development data [SDSV].

2.1.3.3 Smoothing

The building of an n-gram language model requires a large training data that does not include all possible n-grams. But some n-grams which are irrelevant and meaningless can appear at least once in the training data. In order to avoid zero probability problem, the unseen n-gram in training data, different smoothing technique are used [KN95].

"Whenever data sparsity is an issue, smoothing can help performance, and data sparsity is almost always an issue in statistical modeling. In the extreme case where there is so

much training data that all parameters can be accurately trained without smoothing, one can almost always expand the model, such as by moving to a higher n-gram model, to achieve improved performance. With more parameters data sparsity becomes an issue again, but with proper smoothing the models are usually more accurate than the original models. Thus, no matter how much data one has, smoothing can almost always help performace, and for a relatively small effort." Chen and Goodman (1998)

Laplace or add one smoothing (e.g Add-1-Smoothing) is the simplest smoothing technique which is used for the first time by Lidstone and Jeffays [Lid20]. By this method adding the count of one for all possible n-grams as seen in Equation 2.11.

$$\mathbf{P_{laplace}}(\mathbf{w_i}|\mathbf{w_{i-1}}...\mathbf{w_{i-1+n}}) = \frac{\mathbf{C}(\mathbf{w_{i-1+n}}...\mathbf{w_{i-1}}\mathbf{w_i}) + \mathbf{1}}{\mathbf{C}(\mathbf{w_{i-n+1}}...\mathbf{w_{i-1}}) + \mathbf{V}} \tag{2.11}$$

where V is the total number of possible n-1-grams . Add-1-Smoothing is problematic because of the moving of mass probabilities. To solve this problem we have to use Add-δ-Smoothing, where δ is a smaller fractional mass.

$$\mathbf{P_{add\delta}}(\mathbf{w_i}|\mathbf{w_{i-1}}) = \frac{\mathbf{C}(\mathbf{w_{i-1}}\mathbf{w_i}) + \delta}{\mathbf{C}(\mathbf{w_{i-1}}) + \mathbf{V}\delta} \tag{2.12}$$

The problem of this technique is how to choose a good value for δ?.

Discounting, backing-off and interpolation are different strategies and advanced techniques used to improve smoothing for language models [Kat87] [GNW95] [NEK94] [NMW97]. Discounting techniques means to subtract a fixed number from each n-gram count and to distribute it to the unseen or not occur frequently n-grams. In equation 2.13 an absolute discounting method is illustrated.

$$\boldsymbol{P_{abs}}(\boldsymbol{w_i}|\boldsymbol{w_{i-1}}) = \begin{cases} \frac{\boldsymbol{max\{C(w_{i-1}w_i)-D,0\}}}{\boldsymbol{C(w_{i-1})}} & \boldsymbol{if C(w_{i-1}w_i) > 0} \\ \boldsymbol{\lambda(w_{i-1})P_{abs}(w_i)} & \boldsymbol{otherwise} \end{cases} \tag{2.13}$$

Another example for discounting is Good-Turing estimate which is based on the estimation of probability of n-grams which occur r times with the probability of n-grams which occur r + 1 times. $\mathbf{r^*} = (\mathbf{r}+\mathbf{1})\frac{\mathbf{N_{r+1}}}{\mathbf{N_r}}$: where N_r (respectively N_{r+1}) is the number of n-grams that occur r (respectively r+1) times and Discount $\mathbf{d_r} \approx \frac{\mathbf{r^*}}{\mathbf{r}}$.

Another way to smooth the probability distributions of the n-grams is the back-off smoothing techniques like Katz Smoothing. The katz smoothing as presented in equation 2.14 use the Good-Turing discounting. The main idea is to use the lower order n-1-grams model to estimate the probability of n-grams with zero counts and to discount the n-grams with nonzero counts to increase mass probability for the unseen n-grams [Kat87].

$$\mathbf{P_{katz}}(\mathbf{w_i}|\mathbf{w_{i-1}}) = \begin{cases} \frac{C(w_{i-1}w_i)}{C(w_{i-1})} & if r > k \\ d_r \frac{C(w_{i-1}w_i)}{C(w_{i-1})} & if k \geqslant r > 0 \\ \alpha(w_{i-1})P_{katz}(w_i) & if r = 0 \end{cases} \tag{2.14}$$

Chen and Goodman [CG96] have evaluated and compared the different existing smoothing techniques for statistical language models. They judge the dominance of the Katz algorithm for a given large training data and the advantage of the Jelinek-Mercer for a

few training data. Moreover they conclude that the most performed algorithm is the modified Kneser-Ney.

Another technique to smooth a language model is the linear interpolation. It is irrelevant to have seen n-grams in the training data or not. The linear interpolation differs from backoff in that it use always information from lower order n-grams even if the n-grams with non-zero counts. The recursive definition is represented as follows [CG96]:

$$\mathbf{P_{interp}}(\mathbf{w_i}|\mathbf{w_{i-n+1}^{i-1}}) = \lambda_{\mathbf{i-n+1}}^{\mathbf{i-1}}\mathbf{P_{ML}}(\mathbf{w_i}|\mathbf{w_{i-n+1}^{i-1}}) + (\mathbf{1} - \lambda_{\mathbf{i-n+1}}^{\mathbf{i-1}})\mathbf{P_{interp}}(\mathbf{w_i}|\mathbf{w_{i-n+2}^{i-1}}) \tag{2.15}$$

where $\mathbf{P_{ML}}(\mathbf{w_i}|\mathbf{w_{i-n+1}^{i-1}})$ is the n-gram maximum likelihood model and $\lambda_{\mathbf{i-n+1}}^{\mathbf{i-1}}$ is the interpolation weight.

2.1.4 Decoder

As shown in figure 2.1 the decoder combines the language model which provides P(W) and the acoustic model which provides p(X|W). The role of the decoder is to find for a given feature sequence X the word sequence W which maximizes the probability P(W|X) to solve the equation 2.2.

A sequence of states in an HMM represents the word sequence W. The total amount of HMM state sequences defined the search space. A typical search spaces may have half million HMM states sequences. A given 12 words per sentence and a vocabulary of 70,000 words have millions possible word sequences. The critical issue is impossible to compute the most likely sequence of words by evaluating the likelihoods of all possible sequences. Therefore an intelligent algorithm that scans the search space and finds the best hypothesis is needed [Sch12].

The search can be organized in two ways namely depth-first search or breadth-first search. While the depth-first algorithm aims to follow the most promising hypothesis until the end of the speech is reached, the breadth-first approach evaluates all hypotheses in parallel [You96]. Instances of the depth-first search or stack decoding algorithms are the Dijkstra [Dij59] and the A* algorithm [Jel69] [Pau91].

The A* search use a heuristic function to expand the node first which gives the best promise that leads to the best path to the goal. According to [You96] Breadth-first decoding is more frequently referred to as Viterbi decoding.

Where the search space is huge in typical LVCSR tasks, pruning techniques is needed as an optimization method to throw away the unpromising parts of the search space. The beam search approaches which use the pruning technique are particularly effective for LVCSR [NHUTO92].

2.2 Tools

The tools used are the SRI Language Model toolkit which is responsible for building language model. JANUS is used to recognize speech and Sequitur G2P to generate pronunciations of new words. Hyphen and Morfessor to segment words into syllables.

2.2.1 SRI Language Model Toolkit

The SRI Language Modeling Toolkit (SRILM) [S+02], developed by the SRI Speech Technology and Research Laboratory (STAR Lab) is a collection of C++ libraries, executable programs, and helper scripts designed to allow both production of and experimentation with statistical language models for speech recognition and other applications.

SRILM is freely available for research purposes. The toolkit supports creation and evaluation of a variety of language model types based on N-gram statistics, as well as several related tasks, such as statistical tagging and manipulation of N-best lists and word lattices.

The main tasks of SRILM are to estimate and evaluate the statistical language models for speech recognition. Estimation is to create a languages model from training data. Evaluation is to compute the probability of a test corpus, conventionally expressed as the test set perplexity. To perform these tasks, two purposes called ngram-count and ngram are included in SRILM toolkit [S+02].

The function of ngram-count is to estimate n-gram language models after the generation and manipulation of the ngram counts. It is better for text data to count how often words and word sequences occur in. The resulting counts are used to build language model. More options are available by SRILM to adjust it. For examples the basis options are:

- -order n to determine the order of the estimated LM. The default order is 3.
- -text textfile to generate N-gram counts from it
- -lm lmfile to estimate a language model from the total counts and to write it to lmfile
- type of discounting algorithm to use such as Good-turing, absolute, Witten-Bell, and modified KneserNey. The default discounting method is Good-turing.

A standard LM hat trigram order, with Good-Turing discounting and Katz backoff for smoothing would be created by:

ngram-count -text Trainingdata -lm lm

The function of ngram is to score sentence, to compute perplexity, to generate sentences, and to interpolate various types of model. It is responsible to evaluate the performance of a resulting language models on a test data by computing the perplexity as follows:

ngram -lm lm -ppl Testdata

Linear interpolation allows the combination of two or more LMs by taking a weighted sum of the probabilities given by the component language models . The optimization of the interpolation weights is doing on the held-out data. Simple and fast to calculate are the advantages of the linear interpolation. The output is a probability estimate due to the probability estimation of the inputs [BK05].

2.2.2 Janus Recognition Toolkit

The Interactive Systems Laboratories at Carnegie Mellon University, USA and Karlsruhe Institute of Technology (KIT), Germany have developed the Janus Recognition Toolkit (JRTK) which is a speech recognition system [LWL+97] [FGH+97]. Known that the aim of the JRTK is to be used in all speech recognition experiments, it has proven efficacy strikingly in handwriting, biosignal, emotion and silent speech recognition. These toolkit consists of a C codebase configurable via TCL/TK scripts based environments allow building different recognizers.

The janus speech recognizer uses the concept of Hidden Markov Models (HMMs) for acoustic modeling [StÃ12]. Scripts are used to control the speech recognition components from codebooks over dictionaries to the decoder and to make the exchange easy. Moreover the object oriented architecture of the recognizer allows re-utilization of the components. In [RW95] details about the train procedure with Janus.

The ibis decoder which is a part of the JRTK is used to decode the test and to develop the set. It is a one pass decoder described in [SMFW01] that uses linguistic context polymorphism. It uses all available language model information. The Ibis decoder compared to the Sclite [Fis06] from NIST speech recognition scoring toolkit is the best way to evaluate the speech recognition systems.

2.2.3 Sequitur G2P

Sequitur G2P is a trainable data-driven Grapheme-to-Phoneme converter developed at RWTH Aachen [BN08]. It is open source software. Sequitur G2P is able to transform each sequence of graphemes which is the basic unit of written speech to sequences of phonemes which is the basic acoustic unit of speech. It utilizes statistical graphone approach (or joint-sequence models) to accomplish this task. The graphone method is applied to the alignment problem. As shown in Figure 2.2.3 [BN03], the pronunciation of "speaking" may be regarded as a sequence of five graphones.

From an existing training dictionary and after nine iterations of model training it is possible to infer pronunciations of new entries without known pronunciations. The output is performed by finding only the most likely pronunciation sequence for each word. Although pronunciation variations also can be generated, we allowed only one pronunciation. Words contain unseen characters in the training having no output. Therefore, it is advised to clean the list of word to gain time. Sequitur has been used in this work for training and to generate the pronunciations of sub-word and full-word vocabulary.

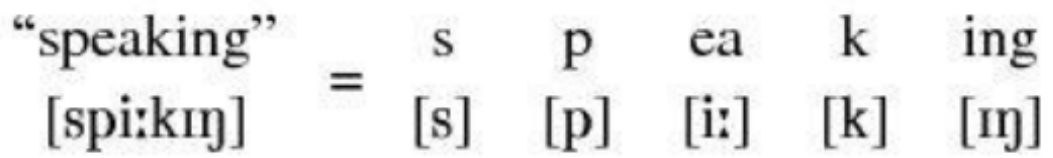

Figure 2.9: Sequitur G2P.

2.2.4 Morfessor

The ultimate goal in the morphological segmentation task is to segment words into morphemes which represent the smallest meaning-carrying units. Morfessor [SVG+14] is an unsupervised morphological segmentation algorithm used to produce a simple morphology of a natural language from a large raw corpus or text data. Morfessor simultaneously builds a morph lexicon and represents the corpus with the induced lexicon using a probabilistic maximum a posteriori mode [CLV06].

Morfessor Baseline is the first version of Morfessor was developed by [CL02], its software implementation, Morfessor 1.0, released by [CL05].

Morfessor 2.0 is a rewrite of the original, widely-used Morfessor 1.0 software, with well documented command-line tools and library interface. It includes algorithmic improvements and new features such as semi-supervised learning, online training, and integrated evaluation code [SVG+14].

Over the past years, Morfessor was used for a wide range of languages and applications. The applications include large vocabulary continuous speech recognition [HCS+06], machine translation [VVCS07], and speech retrieval [ACP+09]. Morfessor is well-suited for languages with concatenate morphology or compound words, and the tested languages include Finnish and Estonian [HPK09], German [EDMSSN10], and Turkish [ACP+09].

2.2.5 Hyphen

According to [Ném06] Hyphen is a high quality hyphenation and justification library based on the TeX hyphenation algorithm. This algorithm [Lia83] was developed in 1983 by Franklin Mark Liang.

"The new hyphenation algorithm is based on the idea of hyphenating and inhibiting pattern. These are simply strings of letters that , when they match in a word, give use information about hyphenation at some point in the pattern. For example "-tion" and "c-c" are good hyphenating patterns. An important feature of this method is that a suitable set of pattern can be extracted automatically from the dictionary."
Franklin Mark Liang (1983)

Peter Novodvorsky from ALTLinux cut hyphenation part from libHnj to use it in OpenOffice.org. The compound word and non-standard hyphenation are supported by László Németh.

In this thesis a German dictionary for hyphenation (hyph_de_DE.dic) is needed as input data and it is based on the converted TeX hyphenation pattern "dehyphn.tex".

3. Related work

In this chapter, we discuss the related work that deals with sub-word models for LVCSR systems. But firstly we have to define some types of sub-word units that are the base of many related work and to talk about the known challenges for sub-word models.

3.1 Sub-Word Units

A sub-word based LM is an estimation of LM through sub-word units or equivalently sub-lexical units. Sub-word units are some fractions of the graphemic word. The size of the vocabulary to be recognized and the sufficient training data for creating effective reference models have a major impact on the choice of the fundamental unit for a recognition task [LJSR89]. The choice of the sub-word type is one of the problems of sub-lexical language modeling. According to [LJ14] phones (which are the basis for writing down a language and the smallest segments of sounds that can be distinguished within words) and multiphone such as syllables, demisyllables, and diphone are possible choices for subword units that can be used to describe a language. The most used type of sublexical units are: morphemes [CPCZ06] [C$^+$06]; [CHK$^+$07] [LPR$^+$03] [XNN$^+$06], syllables [Maj08] [SLE05] [XMZ$^+$96] and graphones based on arbitrary word fragments [BN05] [BN08] [Gal03].

3.1.1 Morpheme

The smallest linguistic component of the word that holds a semantic meaning called morpheme which is one possible type of sub unit Full-words are used to generate morphemes by applying morphological decomposition based on supervised or unsupervised approaches. [MSSN13] Linguistic knowledge is required for the supervised approaches. Therefore, supervised approach is a knowledge-driven approach, while the unsupervised approach is statistical data-driven approaches. Language independent and their applicability to any language are the most important characteristics for unsupervised approach [Mou14].

Generally, morphemes for German, Polish and Turkish LVCSR experiments are generated via unsupervised approaches implemented in a tool called Morfessor. In Section 2.2.4, we describe Morfessor in more detail.

3.1.2 Syllable

A syllable is another type of sub-word unit and is composed of one or more written letters representing a unit of speech [Mou14]. It can also be known as a component of phonological words. The syllables represent a set of written sub-words which can be used for sub-word based language modeling although they are always linked to its pronunciation. Normally, a syllable is made up at least of a central element (nucleus) that can either be a vowel or a diphthong [Mou14]. Consonant clusters can surround the nucleus and must satisfy the phonotactic restrictions to form a valid syllable [KJ96]. In many languages, syllabification which means to divide a word into syllables need linguistic and phonetic rules to achieve it [HM14].

3.1.3 Graphones

A different type of unit is the graphone which is formed by joining together the graphemic sub-word with its context dependent pronunciation. Therefore, a graphone is a combination of two parts which are a graphemic part and a phonemic part.

In LMs it is advised to use graphone to enable the capture of different context dependent pronunciations of sub-words on the LM niveau rather than the lexical niveau. This is an implicit combination of pronunciation model and language model in one common distribution [Mou14].

The problem of high OOV rates can be solved with this approach. The consolidation of the traditional word model with a specialized graphone-based model is dedicated for modeling OOV words. This OOV modeling has the objective to be capable to write out new words as sequences of graphones. Habitually, the presence of the OOV words help to allocate the neighboring words causing the mis-recognition of in-vocabulary words [Mou14].

Beacause each OOV words causes 1.5 to 2 errors rate in [BN05], the successful recognition of OOV words have a positive impact on the recognition of the neighboring words. Usually, the type of graphemic part determines the type of the graphone [BN05] [Gal03], have used only fragment-based graphones, where the graphemic parts are just arbitrary fragments take into consideration length constraints but not linguistic considerations. [BN08] describe that the grapheme-to-phoneme (G2P) conversion model is the base to choose the set of graphones. In Section 2.2.3, we describe Sequitur G2P in more details.

3.1.4 Challenges for Sub-Word models

Two main challenges for sub-word models are: pronunciation variability and data sparseness.

3.1.4.1 Pronunciation Variability

[WTHSS96] and [SOA09] concluded that in conversational speech the spoken words are often pronounced differently from their dictionary pronunciation.

This variability is one of the main challenges facing speech recognition discussed in [OSS05]. [ADL99] describe many reasons (factors) for this pronunciation variability such as the degree of formality of the situation, the relationship and age difference between the speaker and the listener which his language competency is taken into consideration with the background noise.

Substitution of one sound to another can caused variation by context dependent phones and Gaussian mixtures. But by [JWB+01], evident deletion of sound are badly. According to [WTHSS96] speaking style, which causes great pronunciation variability, is an important factor to determine the performance of LVCSR system.

According to [FL99] different words may be pronounced canonical or non-canonical which is predominately not recognized. An example in [LFLM12] in German, "haben wir" (we have) is canonically pronounced [h a: b @ n v i:6] (using the SAMPA international transcription alphabet), but can be pronounced as [h a m a] or [h a m v a] in colloquial speech. Another examples in [ADdMAL05] occur in French; voulait (wanted): [v u l E] $\rightarrow$ [v l E], c'est $\grave{a}$ (that is): [s E t a] $\rightarrow$ [s t a].

3.1.4.2 Data Sparsity

Another challenge for a given training data is the number of sub-word units which can reach a few thousands of triphone units in a typical language. In this case and for insufficient resources such as audio data and dictionary it is difficult to train conventional models for languages or dialects. Therefore it is advised to investigate models based on units that are more language-independent and robust to data sparseness.

In another word, if sub-word occur most frequently than the full-words in training data can help to reduce the effect of data sparsity. The two challenges of sub-word modeling help the speech recognition to progress only in restricted application and large resources languages. Therefore speech recognition is not used for unrestricted applications, such as court room transcription, closed captioning and freestyle dialogue systems [LFLM12].

3.2 Literature Review

[EDMSSN10] investigate the use of sub-lexical LMs for German large vocabulary and continuous speech recognition (LVCSR). They compare three approaches for word decomposition which are supervised, unsupervised word decomposition and the graphone-based decomposition. Moreover, they conclude that the best approach is to use a vocabulary of fragments generated by unsupervised methods, along with some fraction of full-words (around 5k).

Linguistic knowledge is required for supervised approaches. For example in [ADA00] a set of about 340 rules has been manually developed for splitting compound German words. [BFRB96] use a hand corrected lexicon for recognition, where compound words are manually decomposed. Lexical and syntactic knowledge in [EDGR+09] [KK01] are the base of other supervised approaches rely on morphological analysis.

Supervised methods have a positive impact on the performance of the recognizer, but they require labor-intensive work. On the contrary the unsupervised approaches, which are data-driven statistical-based approaches do not need any linguistic knowledge and can be applied to any language. In [AD03] a set of 800k decomposition rules are automatically extracted.

The minimum description length principle (MDL) [CHK+07] and compound splitting algorithm [OVHDJ03] [LWKR00] are the base of other unsupervised methods. In [CHK+07] the development of the compound splitting algorithm is based on sorting, word length, and word frequency information. While in 8 the splitting of compound word depends on the statistical relevance of the resulting constituents.

In [SMSN11a] the use of morpheme and syllable based units is investigated for building sub-lexical LMs for LVCSR of Polish. Here, morphemes and syllables are combined with their pronunciations and are the base of a different type of sub-lexical units. They build LM based on graphone. They used the text corpora to select vocabulary (N most frequent words) and to estimate back-off N-gram LMs by the SRILM toolkit. Moreover, they concluded that the best results is morphemic graphones with a vocabulary of 70k full-words plus 277k graphones.

[SMSN11b] presents the use of hybrid lexicons and LMs based on three mixed types of sub-lexical units for building an open vocabulary LVCSR system for German language. For the most frequent in-vocabulary words, normal full-words are used. While, for less frequent in-vocabulary words, graphemic morphemes or syllables are used. According to [SMSN11b] the use of morphemic sub-words outperforms the use of syllabic sub-words for German language. Due to the high length of compound word in German the number of syllables per word is relatively much great than the number of morphemes

In [KK01], the authors propose four different approaches to segment words into shorter fragments. Depending on the needed target function (OOV-rate, WER, LER) one of the segmentation strategies comes off as winner.

[VKV13] presented a novel algorithm, called Greedy 1-Grams (G1G), which learns a subword vocabulary based on unigram likelihood. It provided the best performing sub-word vocabulary for a Finnish LVCSR task.

[LFLM12] reviews past, present, and emerging approaches to sub-word modeling. In order to make clean comparisons between many approaches, the review uses the unifying language of graphical models. They have motivated the need for breaking up words into sub-word units and surveyed some of the ways in which the research community has attempted to address the resulting challenges, including traditional phone-based models and less traditional models using acoustic units or subphonetic features.

In [MSD+12], the authors propose a simple approach to learn the subword units from the data. This approach is based on very simple rules and frequency of occurrence of these units in the training data. The idea is to keep the most frequent words and split all the meaning words into syllable and to keep the most frequent syllables plus words and split all remaining tokens into individual characters. This approach guarantees that all infrequent words can be spelled into characters or syllables.

[SSG10] use statistical (no linguistic knowledge is required) and grammatical word splitting approaches for Turkish language. Even in noisy environments [AT08] using syllables acoustic units improve the performance of ASR systems of Arabic spoken proverbs.

[Par11] propose a probabilistic model to learn sub-word units for hybrid speech recognizers by segmenting a text corpus while exploiting side information.

[LJSR89] discuss the use of three types of fundamental units. Namely whole word units, phoneme-like units and acoustic subword segment units, the last two represent the subword units. Linguistic definitions define the phoneme-like units, while acoustic signal realizations specify absolutely the acoustic segment units. According to [LJSR89] and [T.S88] by the acoustic segment units there is no mismatch problem as encountered with the phoneme-like units or linguistic sub-word.

4. Properties of German Language

In this thesis all the experiments are based on German language which represent a good example of morphological rich languages. A brief overview of the linguistic characteristics of German language are presented in the next section.

4.1 German Morphological Rich Language

As presented in figure 4.1, German is one of a number of Germanic languages, a family which also includes Dutch, Norwegian, English, Danish and Swedish. It is one of the world's major languages and the most widely spoken first language in the European Union. Globally, German is spoken by approximately 120 million native speakers and also by about 80 million non-native speakers. In Germany, Austria, and Liechtenstein is German the only official language and one of the official languages of Switzerland, Luxembourg, and Belgium. German is the second most commonly used scientific language and the third largest contributor to research and development. It is also a dominant language in business, culture, history, literature, philosophy and theology [Wik15a]. Germany is ranked number 5 in terms of annual publication of new books. One tenth of all books (including e-books) in the world are published in German. German is also after English and Russian, the third most used language used by websites [w3t15].

German is a highly inflected language and contains a large vocabulary. Due to the inflection phenomenon, a large number of words can be derived from the same root. A root, or root word, is defined as a word that does not have a prefix (in front of the word) or a suffix (at the end of a word). The root word called base word is the primary lexical unit of a word, and of a word family, which carries the most significant aspects of semantic content and cannot be reduced into smaller constituents [Wik15b]. For example from the root word "'fahr"' "'driving"' can be formulated "'fahren"' "'drive"', "'Fahrer"' "'driver"', "'Fahrt"' "'ride"', "'Nachfahr"' "'descendant"', "'Fahrrad"' "'bicycle"', "'fahrend"' "'driving"', etc.

4.1.1 Prefix

According to [Mih11] a prefix is an affix which is placed before the stem of a word. Adding it to the beginning of one word changes it into another word. Also, the using of

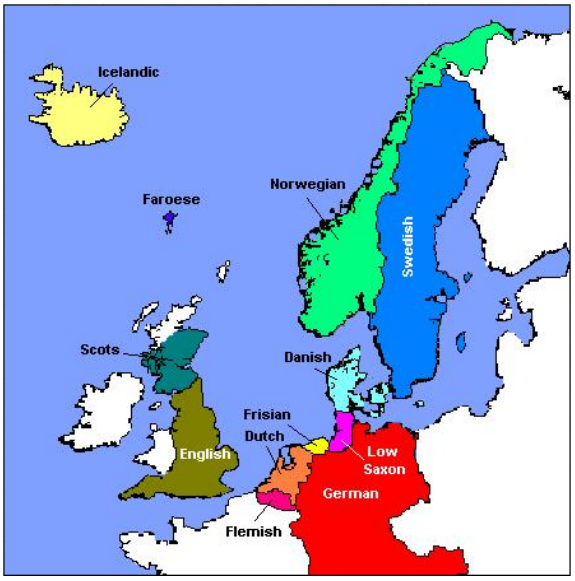

Figure 4.1: The Germanic languages today.

prefixes can change the meaning of German verbs. Most of them are separable prefixes and are derived from prepositions. As their name would imply, separable prefixes can be detached [Dar15b]. That means to split off the separable prefix and to move it after the verb or to the end of the sentence. For example, "mitgehen" - "to go along" which can be split like in "Gehen Sie mit?" - "Are you going with?" [Dar15b]. Another example is "durchfahren". While the prefix "durch-" can have various definitions, here it takes on the meaning of continuation through to an end. Hence 'durchfahren" means: to pass through; to go non-stop [Dar15b]. The table 4.1 contains some examples of the separable prefixes.

ab-	durch-	aus-	auseinander-	dabei-
um-	vor-	weg-	wieder-	zu-
an-	auf-	bei-	da-	dazwischen-

Table 4.1: Some German separable prefixes.

The table 4.2 contains the different possible inseparable prefixes of German language. As the term "inseparable" indicates, the inseparable prefix, in contrast to a separable one, remains attached to the stem in all forms of the conjugation, including the finite form [Dar15b]. For example the prefix "be-" in word "besprechen" - "to discuss": "Wir besprechen die Situation" - " We're discussing the situation".

be-	emp-	ent-	er-
ge-	hinter-	miss-	wider-
ver-		zer-	

Table 4.2: German inseparable prefixes.

4.1.2 Suffix

A suffix is a group of letters placed after the root of a word. The German language contains about 113 suffixes. German often employs suffixes to add meaning or to produce other

parts of speech. In one hand some of suffixes presented in table 4.3 can form Nouns. For example the suffix "-ung" is frequently used to create a noun die "Befreiung" - "liberation" by attaching it to a verb stem "befreien" [Dar15c]. In other hand some of suffixes in table 4.4 can form adjectives and adverbs. For example the suffix "-bar" can be affixed to nouns or verbs to denote "-ability" or a possession of the implied quality: "dankbar" - "grateful", "lesbar" - "legible" [Dar15c].

-ant	-art	-chen	-e	-ent
-zeug	-ung	-keit	-heit	-er
-ling	-or	-lein	-schaft	-tät

Table 4.3: Some Suffixes for Forming Nouns.

-arm	-artig	-fach	-frei	-haft
-mal	-los	-lich	-isch	-ig
-voll	-sam	-wert	-würdig	-er

Table 4.4: Some Suffixes for Forming Adjectives and Adverbs.

4.1.3 Compound Word

Mark Twain said *"the compound words the German language uses to capture precise or complex meanings, which are a cause of irritation for novices and a delight for those who manage to master the tongue".*

German offers the possibility of combining of words. Compound words are formed when two or more stem words are put together to form a new word with a new meaning. Compound words fall within three categories and it is not unusual to find the same word in more than one group. Here are the three types of compound words [k1215]:

- Closed compound words presented in table 4.5 are formed when two unique words are joined together.
- Open compound words illustrated in table 4.6 have a space between the words but when they are read together a new meaning is formed.
- Hyphenated compound words presented in table 4.7 are connected by a hyphen. The sign (-) used to join words to indicate that they have a combined meaning.

	Noun	Verb	Adjective	adverb	Preposition
Noun	Wort+bildung	seil+tanzen	blitz+schnell	fluss+abwärts	
Verb	Koch+topf	dreh+bohren	klopf+fest	Tauge+nichts	Reiss+aus
Adjective	Blau+helm	rein+waschen	hell+gelb	rund+weg	rund+um
Adverb	Wieder+wahl	davon+laufen	immer+grün	immer+fort	aussen+vor
Preposition	Gegen+satz	wider+sprechen	vor+laut	vor+weg	neben+an

Table 4.5: Some closed compound words.
[Rei05]

Adjective + Verb	Noun + Verb	Verb + Verb
dabei sein	Rad fahren	fahren lassen
gesund sein	Schuld haben	kennen lernen
laut reden	Tennis spielen	sitzen bleiben
sauber schreiben	Staub saugen	spazieren gehen
frei sprechen	Recht bekommen	liegen lassen

Table 4.6: Some open compound words.

Eurozonen-Ländern	UN-Generalsekretär	Politik-Arbitrage
Online-Ressource	deutsch-französischen	Kaffee-Ersatz
medizinisch-technische	geistig-kulturelle	Hoch-Zeit
Internet-Zensur	EU-Haushalts	Mehrzweck-Küchenmaschine
Schwimm-Meisterschaft	Lotto-Annahmestelle	Umsatzsteuer-Tabelle

Table 4.7: Some hyphenated compound words.

The closed compound words in table 4.5 have no connect element between the determiner and the primary word. But in some other cases is needed to use the connect element such as "-e-" in "Wartezimmer" - "waiting room", "-en" in "Gedankenfreiheit" - "freedom of thought" and "-s-" in "Staatspolizei" - "state police" to form the resulted closed compound words [Dar15a].

German language have a large quantity of closed compound words, especially nouns. The most compound verbs are a combination of root word and a prefix or suffix.

Moreover, the German language is known for its extremely long compound nouns. The Duden dictionary includes a 67 letter compound noun: "Grundstücksverkehrsgenehmigungszuständigkeitsübertragungsverordnung", which means "Land transport permit transfer of competence Regulation".

In this thesis only the hyphenated and closed compound words are investigated.

5. Experimental Setup

In this chapter we define our baseline sub-word language models for German LVCSR. Then we define the different data resources and scripts used for our experiments. Some experiments have the same implementation methods and belong to one of the proposed three scenarios.

5.1 Expanded Baseline sub-word German ASR system

The goal of this thesis is to buil a sub-word language models for German LVCSR. Therefore, our baseline is based on an expanded sub-word German LVCSR system, with 19,2 WER, done by the Institute for Anthropomatics in KIT [KHM+].

The data sources of the expanded baseline system are:

- 180 hours of Quaero training data from 2009 to 2012.
- 24 hours of broadcast news data.
- 160 audio from the archive of parliament of the state of Baden-Württemberg, Germany.

According to [KHM+] the Quaero training data is manually transcribed. Moreover using the "dev2014" for test. The segmentation of audio data is automatic based on the SVM segmentation approach.
The expanded baseline system use context-dependent quinphones with three states per phoneme and a left-to-right HMM topology without skip states. The German acoustic models use 6000 distributions and codebooks [KHM+]. An initial pronunciation dictionary based on the Verbmobil Phones is used. The building of the language models is based on the SRILM toolkit with modified Kneser-Ney smoothing and the top 300k German words [KHM+].

In this thesis we use the same expended baseline acoustic model and test speech data. And we use a different pronunciation dictionary and language model for the decoding task.

5.2 Data Resources

5.2.1 Corpus Data

To have the possibility to compare our results to the result of the expanded baseline ASR system, we have to use the same text corpora. Therefore, all experiments in this thesis are based on 10 texts corpora chosen from 28 texts of the expanded baseline ASR system. As shown in table 5.1 if we use these 10 texts to build the language model and take the same dictionary and acoustic model of the expanded baseline ASR system, the word error rate (WER) is 22,5%.

ASR system	Sub-word Expanded Baseline (Exp. B.)	Our Sub-word Baseline
Dictionary	Exp. B. Dict.	Exp. B. Dict.
AM	Exp. B. AM	Exp. B. AM
LM	Exp. B. 28 Texts LM	10 Texts LM
WER	19,2%	22,5 %

Table 5.1: Expanded Baseline and our Baseline Sub-word ASR system.

The table 5.2 presented the characteristics of these different 10 texts. These characteristics are the size of the text, the number of full-words and the number of hyphenated composed words.

	Size	# Full-words	# Hyphenated C. Words
Text 1	331M	47,725k	156k
Text 2	32M	4,498k	23k
Text 3	216M	32,176k	367k
Text 4	736M	109,716k	1,236k
Text 5	689M	102,797k	1,238k
Text 6	312M	46,477k	560k
Text 7	1,7G	256,671k	3,021k
Text 8	3,7G	553,790k	6,280k
Text 9	2,2G	325,516k	3,871k
Text 10	37M	5,545k	14k

Table 5.2: The 10 used Texts.

5.2.2 Dictionary

In our different experiments we use the same baseline acoustic model, but we should build the language model and the dictionary. Therefore, the Sequitur G2P tool is used to generate our dictionary.

g2p.py –model model-9 –apply dict-vocab

Previously we have to create the models by using the baseline dictionary as "train.lex". In this thesis we use the model-9 to extend the dictionary. Here the different procedure needed to train the models:

g2p.py –train train.lex –devel 5% –write-model model-1
g2p.py –model model-1 –train train.lex –devel 5% –write-model model-2
g2p.py –model model-2 –train train.lex –devel 5% –write-model model-3
.
.
.
g2p.py –model model-8 –train train.lex –devel 5% –write-model model-9

5.2.3 Scripts and Tools for Split

In this section we describe the used scripts and tools to decompose a word or a compound words into sub-words. And we explain the the employment of the select vocabulary script and the letter n-gram script.

5.2.3.1 Hyphenation Tool

The figure 5.1 show how we use the Hyphenation tool in this thesis and mainly in Scenario 1 and Scenario 3. The input data are our vocabulary or text corpus and the German dictionary for hyphenation (hyph_de_DE.dic). The output are split words into suitable syllable. For example "bundeskanzleramt" - (Federal Chancellery) is split into"bun=des=kanz=ler=amt"

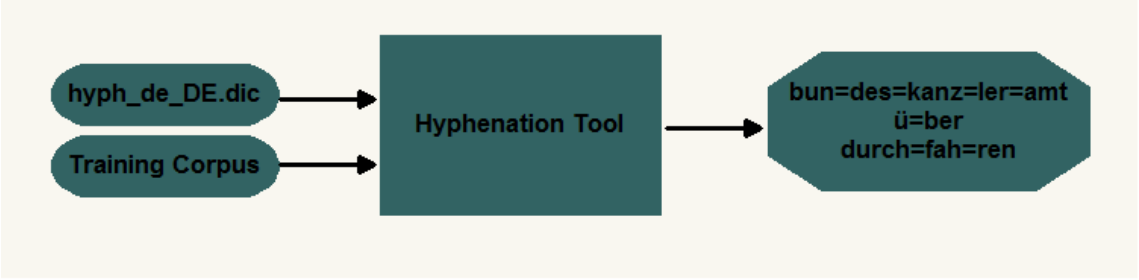

Figure 5.1: hyphenation Tool.

5.2.3.2 Morfessor Tool

As shown in figure 5.2 we use the Morfessor Tool to decompose the word into sub-words.

morfessor -t Split-Vocab -T Text-To-Split –output-format-separator "+ "

Our Morfessor split model use the "Split-Vocab". This vocabulary of distinct words should contain the most frequently words in the training corpus. We do not include less frequent words in order to avoid irregularities. Morfessor is able to decompose all unseen words.

As we said the split by Morfessor is based on the given "Split-Vocab". Therefore, if the "Split-Vocab" not contains the sub-words that could compose a word, the Morfessor split this word into characters.

For example, if we want to split the word "durchfahren" - (drive through) and the "Split-Vocab" contains "durch" and "fahren". The result is "durch+ fahren". Here we attach a "+"

marker to the end of every sub-word to allow for a deterministic recovery to full-words in the recognition output. But if we have "durch", "fahren", "fah" and "ren" as vocabulary in the input "Split-Vocab", the result is "durch+ fah+ ren". The Morfessor tool follows the principle of the minimum description length (MDL). That means that the Morfessor always try to use the smallest given sub-word to decompose a word.

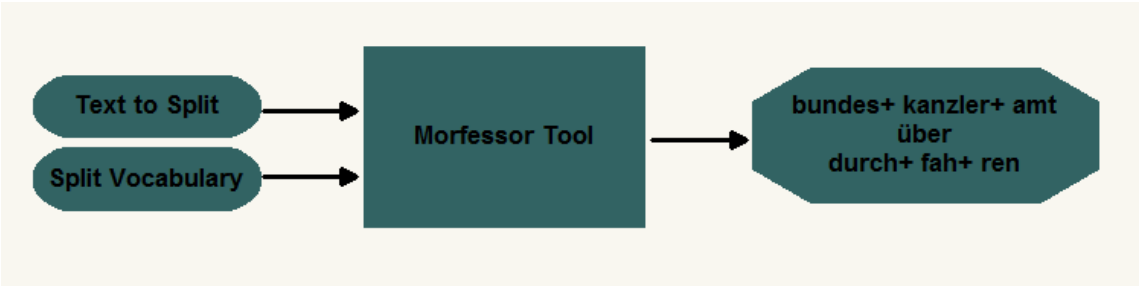

Figure 5.2: Morfessor Tool.

5.2.3.3 Full Compound Split Script

As shown in figure 5.3 we use the Full Compound Split Script to decompose the word into sub-words.

cat Text-To-Split | ./Full-Compound-Split-Script -c –filtermode –vocab Split-Vocab

The input data should contain the "Split-Vocab" and the "Text-To-Split". Always the decomposition of the word depends of the given "Split-Vocab". For example, we want to split the word "Bundeskanzleramt"- (Federal Chancellery). If the "Split-Vocab" contains the vocabulary "bundes", "kanzler", "bundeskanzler" and "amt", then the possible result are "bundeskanzler+ amt" and "bundes+ kanzler+ amt". But the Full-Compound-Split-Script hat a function to choose the best one. The table 5.3 below explains how to choose the best ($178 > 94$), if we have more then one decomposition possibilities.

In other words, when there is more then one syllabification possibilities, the one with the longest syllables is chosen.

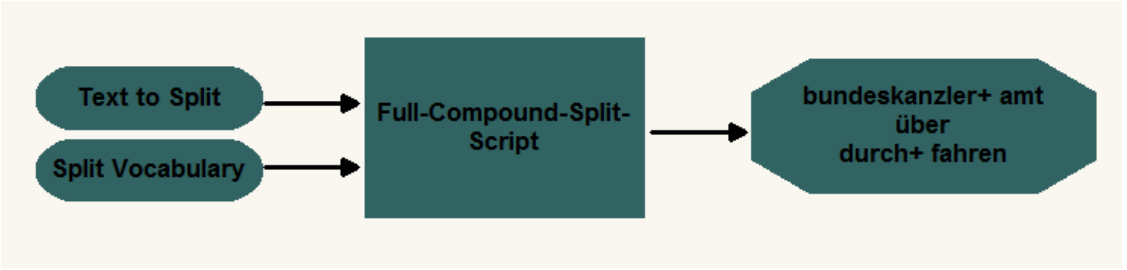

Figure 5.3: Full Compound Split Script.

bundeskanzler+		amt	$13^2 + 3^2 = 178$
13 letters		3 letters	
bundes+	kanzler+	amt	$6^2 + 7^2 + 3^2 = 94$
6 letters	7 letters	3 letters	

Table 5.3: The possibilities to split the word "Bundeskanzleramt" based on the given Split-Vocab.

It is possible that some words will not be decomposed. Because the "Split-Vocab" not includes the sub-words that could compose these words. In this case and different to the Morfessor the "Full-Compound-Split-Script" not decompose words into characters.

5.2.4 Split-Word-Syllable-Character-Script

The figure 5.4 shows the script used in scenario 3 to split the training corpus. The split vocabulary contains top Wk words, top Sk syllables and vocabulary characters. All vocabulary in training corpus will be decompose into syllables and character if it is not include in the top Wk.

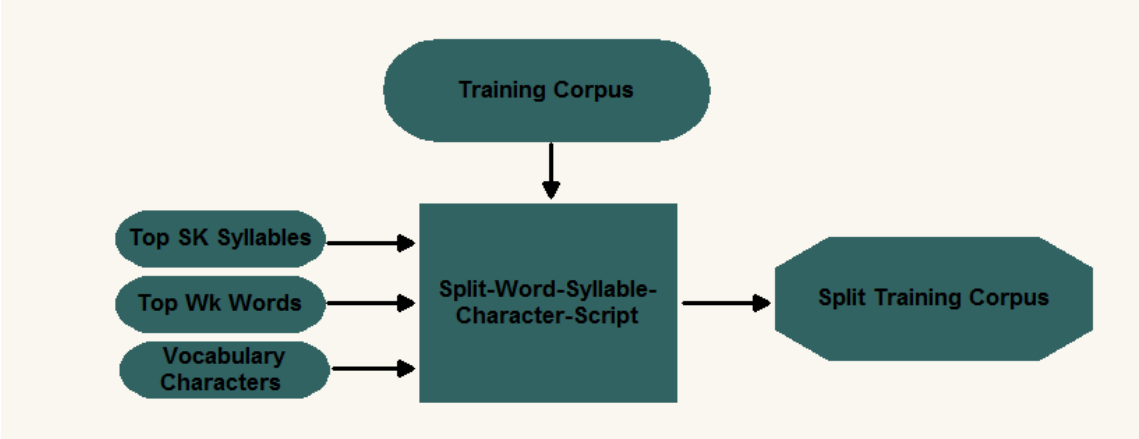

Figure 5.4: Split-Word-Syllable-Character-Script.

5.2.4.1 Select-Vocabulary-Script

The Select-Vocabulary-Script is one of the SRI language Model Toolkit scripts. It selects a maximum-likelihood vocabulary from a mixture of corpora.

Select-Vocabulary-Script -heldout file text 1 text 2 ... text 10

The necessary input data are the held out file and our 10 split texts which are split by using the "Full-Compound-Split-Script". The "Select-Vocabulary-Script" picks a vocabulary from the union of the vocabularies of text 1 through text 10 in order to maximize the likelihood of the heldout file [AV03].
The output as shown in figure 5.5 is the list of words in all of the input corpora together with their weights. This list may subsequently be sorted to put the words in decreasing order of weight [AV03].

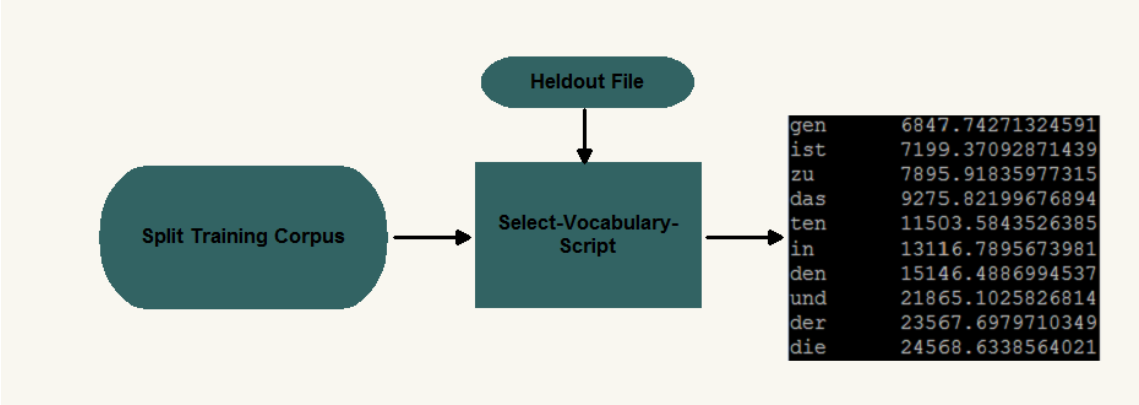

Figure 5.5: Select Vocabulary Script.

5.2.4.2 Letter-N-gram Script

We use the Letter-N-gram Script to split the word into Letter-N-grams character. The size of the Letter-N-grams is from 2 up to 6 Letters. For example the word "dienst" - (service) as input and the output are "di, ie, en, ns, st, die, ien, ens, nst, dien, iens, enst, diens, ienst " and "dienst". The figure 5.6 shows even how to produce Letter-N-grams from the word "straße"- (street).

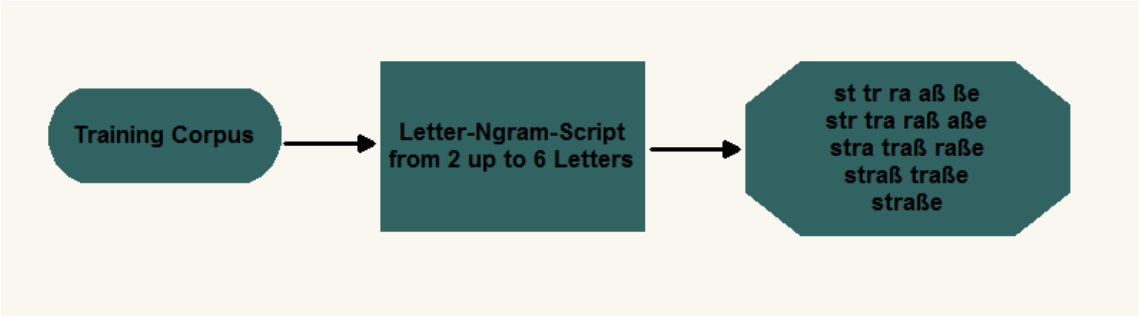

Figure 5.6: Letter-N-gram Script.

5.3 Experiments Scenarios

The different experiments in this thesis have applying one of three possible scenarios. The term "scenario" means that we have always the same experiments implementation and we only change the input data. These three scenarios will be presented with more details. Moreover they share the same expanded baseline ASR acoustic model, the test speech data and the same methods to build the dictionary 5.7 and the language model 5.8.

5.3.1 Building Dictionary

To build the dictionary two input data are needed. These data are the extended baseline dictionary and the dictionary vocabulary. We use the "Search-Vocab-Pronunciation-Script" to search the vocabulary in the extended baseline dictionary. If it exist we take it and if not exist we use the Sequitur G2P to generate word pronunciation.

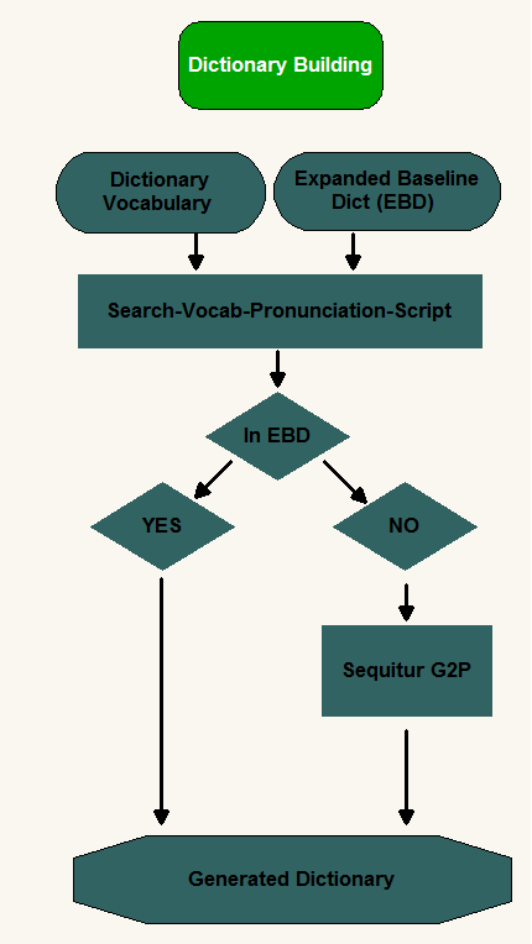

Figure 5.7: Building Dictionary.

5.3.2 Building Language Model

The building of the language model is based on the input vocabulary also called language model vocabulary and the split training corpus. We have to choose the n-gram language model. In the three scenarios the most used n-gram is the 4-gram.

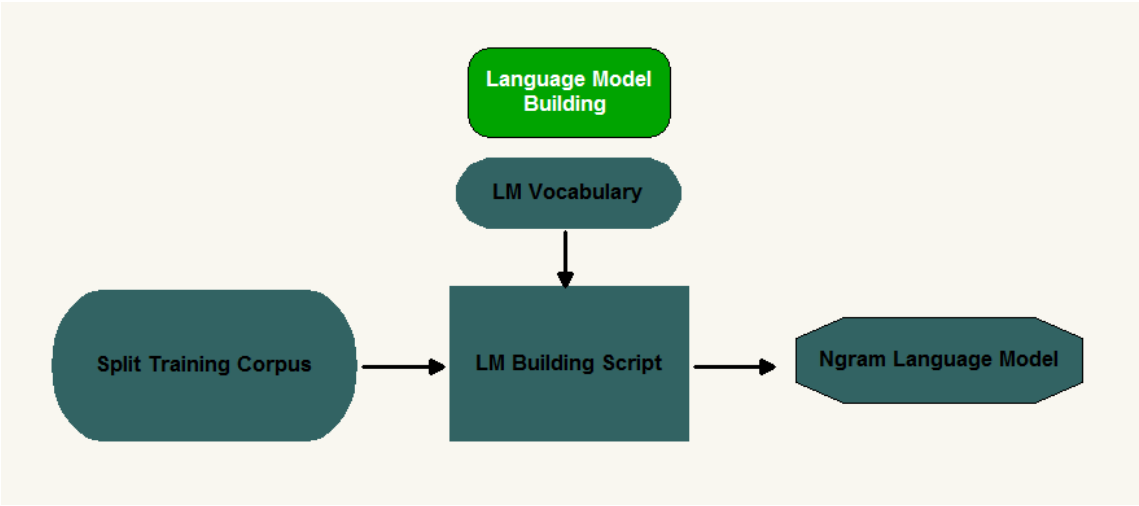

Figure 5.8: Building Language Model.

5.3.3 Scenario 1

The figure 5.9 represents the different stages for scenario 1. The "Training Corpus" contain the 10 training texts. First, we use the "Hyphenation Tool" based on the German dictionary for hyphenation (hyph_de_DE.dic) to split the "Training Corpus". The result is a hyphenated training corpus which contains syllables. These will be sorted by occurrence to take the top 10000 vocabularies as "Split Vocabulary".

Second, the "Split Vocabulary" beside the "Full-Compound-Split Script" are needed to split the "Training Corpus". The most frequently vocabulary will be selected based on the "Select-Vocabulary Script". These vocabularies varied between 10k and 100k are necessary to build the dictionary and the language model.

5.3.4 Scenario 2

The main goal of the scenario 2 is to compare three different experiments based on three different input split vocabulary (Split-Vocab) to split the training corpus. The three input split vocabulary are top 10k full-words, top 10k letter n-grams and top 10k full-words + 10k letter n-grams. The same dictionary will be used by the three experiments and three new different language models must be built.

5.3.4.1 First Approach

The figure 5.10 shows two separately methods to find the most frequently words and letters n-gram based on the "Training Corpus" as input data. On the left, the words are sorted by occurrence. On the right, after the using of the letter n-gram script we have a letter n-gram vocabulary. The length of the letter n-grams varied from 2 up to 6 letters. We take always the most frequent.

Both top Wk words and top LNk letter n-grams represent together the language model and dictionary vocabulary. In this First Approach two split methods Morfessor Tool and "Full-Compound-Split Script" are used to split the training corpus. We want to know if the Morfessor is helpful to improve the WER in the scenario 2.

5.3.4.2 Second Approach

The second approach of scenario 2 is based as shown figure 5.11 of top 10k letter n-grams to split the training corpus. Using the "Full-Compound-Split Script" to build for the second approach its own language model.

5.3.4.3 Third Approach

The third approach differs to the first two approaches by combining the top 10k letter n-grams and the most frequent 10k full-words to create the needed split vocabulary (see figure 5.11). We combine the top Wk full-words and the top LNk latter n-grams to build the language model and the dictionary.

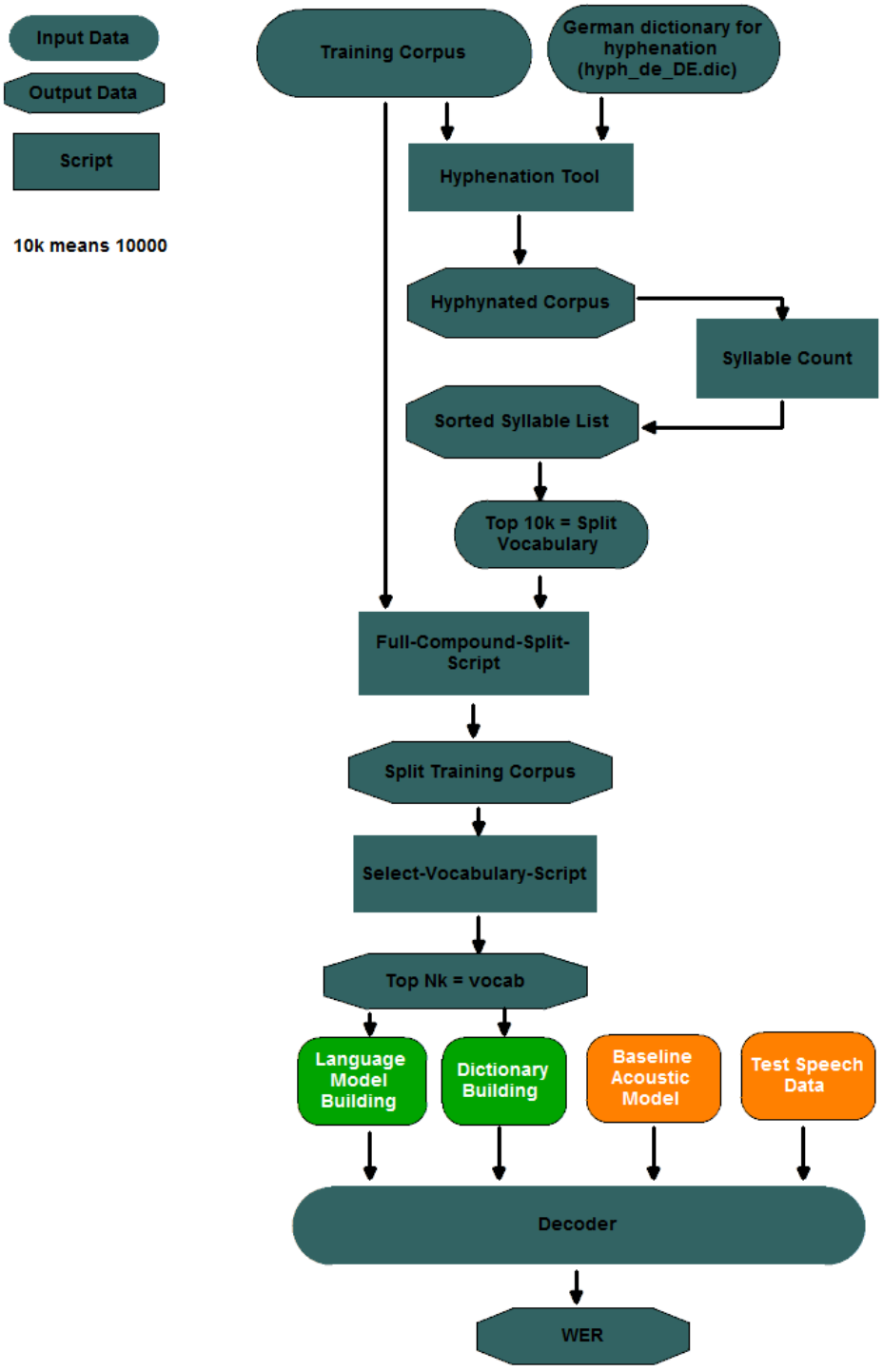

Figure 5.9: The Main Stages of Scenario 1.

5.3.5 Scenario 3

The figure 5.12 shows the scenario 3 which share with the scenario 1 some same implementation phases. The result of the count of vocabulary of the "Training Corpus" is a 9

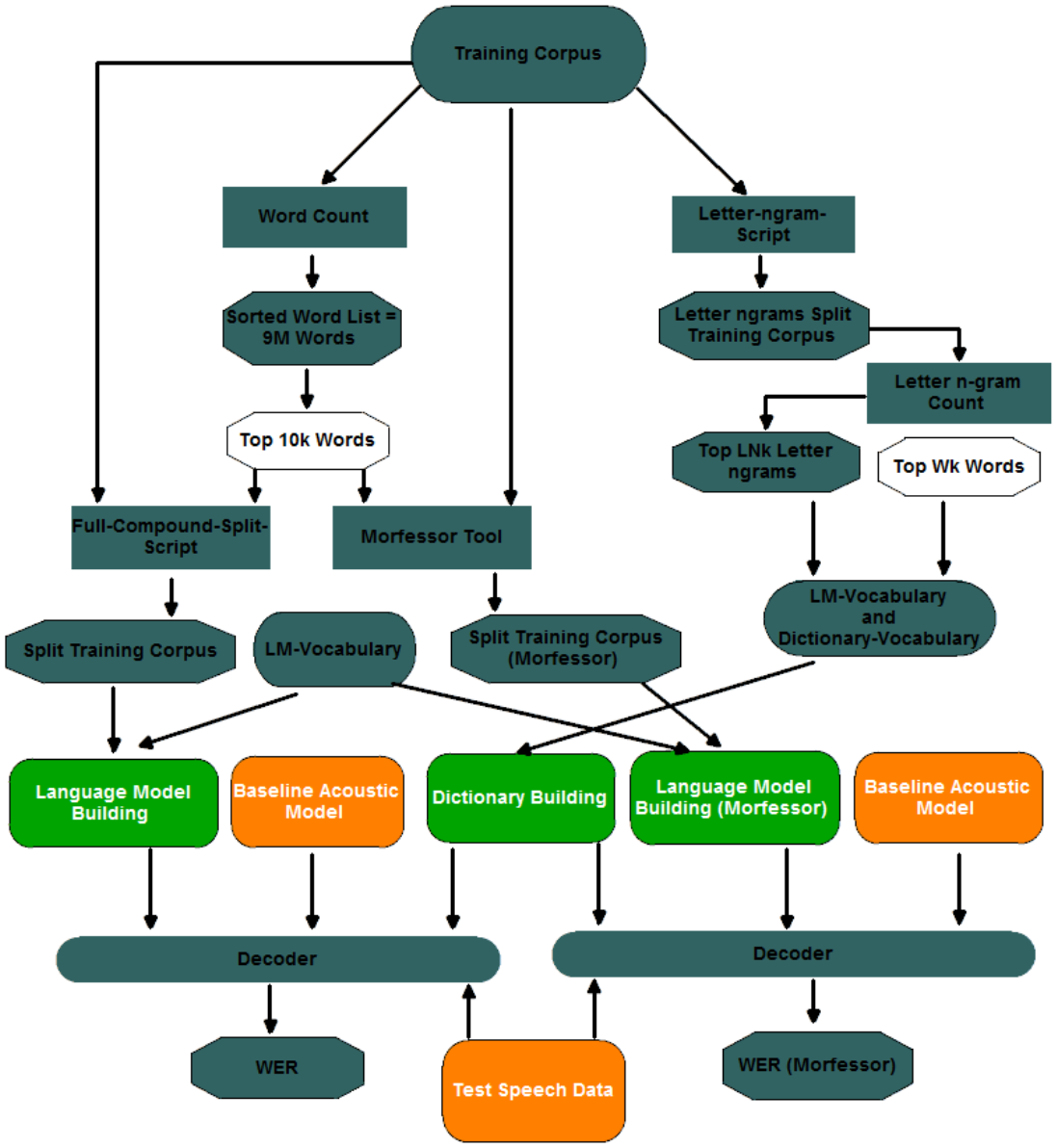

Figure 5.10: The Main Stages of First Approach Scenario 2.

million sorted words list. We use the 60k vocabulary of the extended baseline as the split vocabulary. The training corpus after split will be hyphenated and sorted.

The idea of the scenario 3 is to keep the most frequent words and split all the meaning

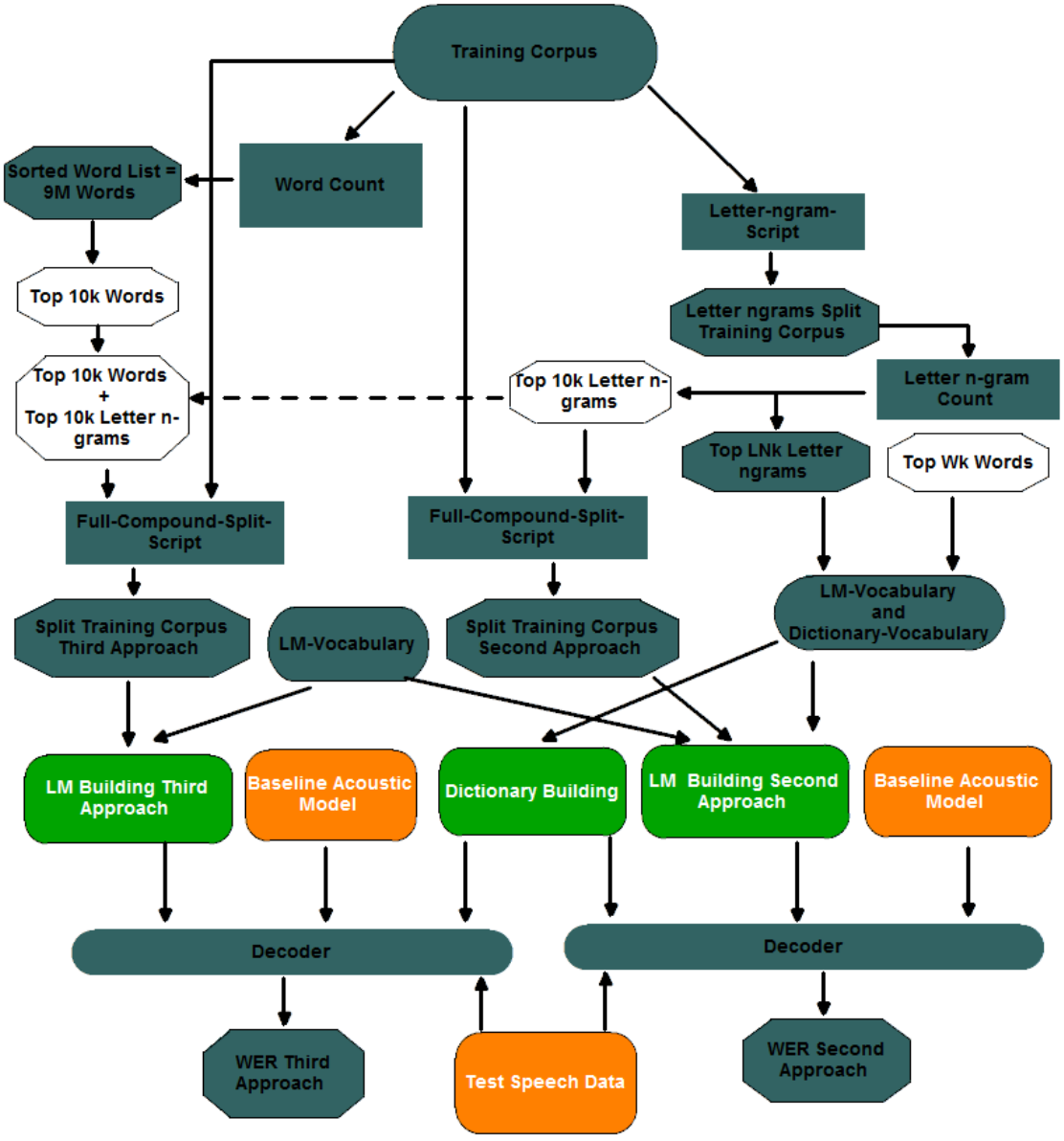

Figure 5.11: The Main Stages of Second and Third Approach Scenario 2.

words into syllable and to keep the most frequent syllables plus words and split all remaining tokens into individual characters. This approach guarantees that all infrequent words can be spelled into syllables or characters.

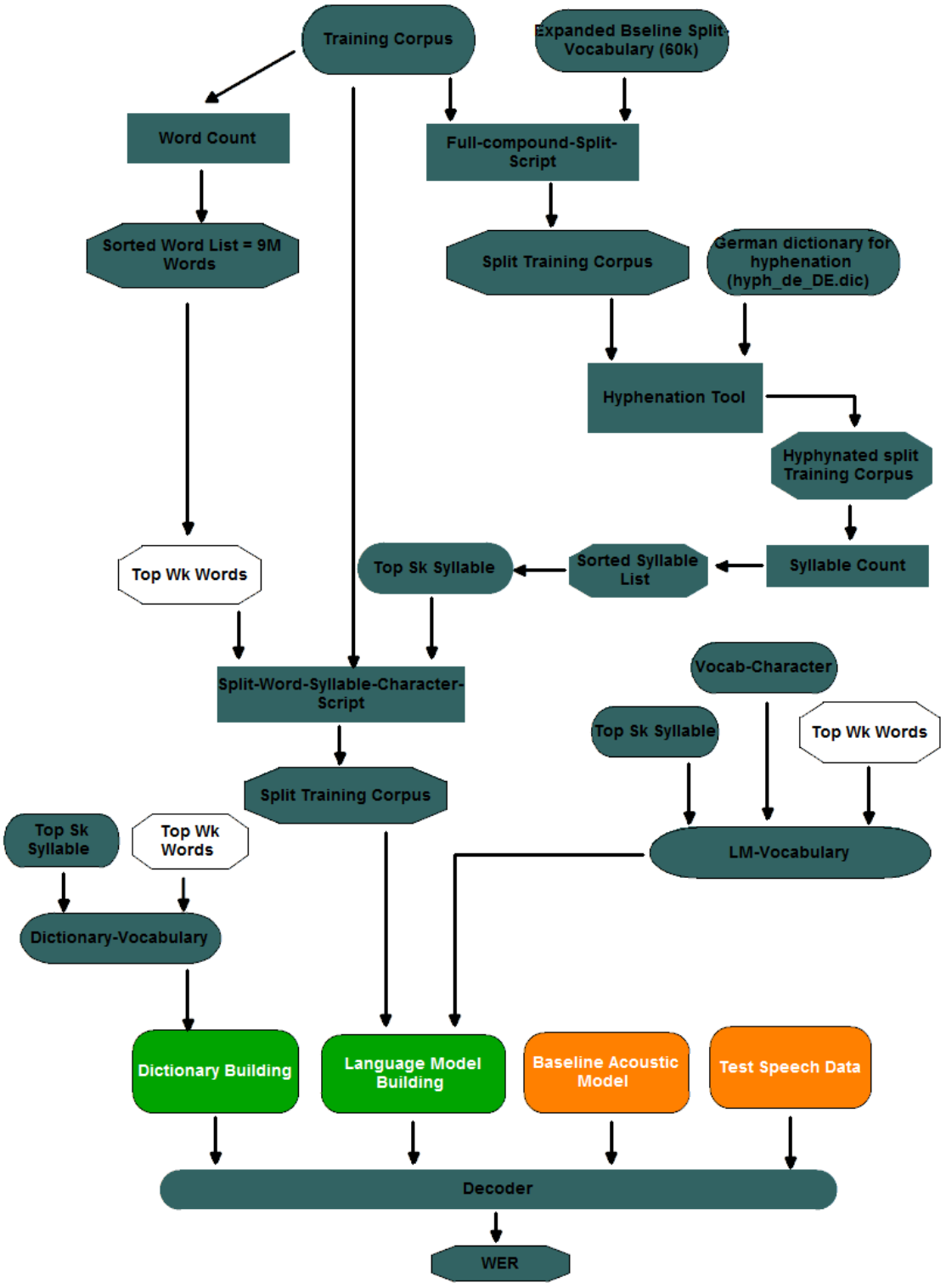

Figure 5.12: The Main Stages of Scenario 3.

What distinguishes the scenario 3 from the other scenarios is combining the top Wk words and the top Sk syllables to form the new split vocabulary. Also the language model vocabulary contains the top Wk words, the top Sk syllables and the vocabulary character. While only the Wk words and the top Sk syllables are needed to have the dictionary vocabulary.

6. Experiment and Evaluation

In this chapter, all the experiments are described and the results are evaluated and analyzed. In each scenario we build an ASR system and calculate the word error rate (WER). Sometimes we find some practical problems that can be solved by theoretical knowledge or change the input data and compare the output. To make mistakes which are in the eyes of some academic intuitive due to their experience and then try to overcome them helps us to understand more the mechanism of the ASR system.

To understand better the three scenarios which were implemented independently from each other we should describe the origin of the input data to the decoder. Since the acoustic model and the test speech data do not vary during our experiments. The main questions that remain are:

- What is the used split vocabulary to split the training corpus?
- What are the used vocabularies to build the LM and the dictionary?

The table 6.1 describes the three scenarios that we considered in our experiments in order to answer the previous questions.

	Split Vocab.	LM Vocab.	Dict Vocab.
Scenario 1	10k TS	Top Nk	Top Nk
1. App of Scenario 2	10k TW	TW + TLN	TW + TLN
2. App of Scenario 2	10k TLN	TW + TLN	TW + TLN
3. App of Scenario 2	10k TW + 10k TLN	TW + TLN	TW + TLN
Scenario 3	TW + TS	TW + TS + Vocab. character	TW + TS

Table 6.1: The needed Vocabulary for all Scenarios.

- Top Nk: the most frequent vocabulary after using the select vocabulary script.
- Top Wk (**TW**): the most frequent words in the training corpus.
- Top Sk (**TS**): the most frequent syllables in the training corpus.
- Top LNk (**TLN**): the most frequent letter n-grams in the training corpus.

6.1 Experiments of Scenario 1

After the split of the corpus data based on the hyphenation script, we use the 10k most frequently syllables as split vocabulary to split the training data. We then use the select vocabulary script to choose the top 100k vocabularies which we use to generate the dictionary and to build the language model.

When building the dictionary, we first search for the dictionary vocabulary in the extended baseline dictionary and adopt them if found. Otherwise we use the Sequitur G2P tool to generate them. The table 6.2 shows a comparison between using only the given dictionary vocabulary and the G2P tool and additionally using the expanded baseline dictionary as a reference. This method has proved to be was a good way 5.7 of building the dictionary and will be adopted in the rest of the scenarios.

	Nur G2P Dictionary	G2P with original Dict.
WER	47.4%	40.9%

Table 6.2: The generation of the Dictionary.

We tried to generate the language model using different n-grams: which are 4, 5 and 6 - gram. The results of Word Error Rate are shown in the table 6.3. We notice that increasing the n-gram does not necessarily improve the WER. We use the 4-gram in the following scenarios as it has the best results in the first scenario.

	4-gram LM	5-gram LM	6-gram LM
WER	40.9%	41%	45.2%

Table 6.3: Results of increased n-gram LM.

6.2 Experiments of Scenario 2

In the second scenario, we used "letter n-grams" instead of syllables. The letter n-grams are brute force split sub-words sorted by occurrence. We split the corpus data into every possible sub-word of 2 to 6 letters and take the most occurring ones.
The reason of to choose 2 letters as minimum length and 6 as maximum length of the letter n-grams is interpreted from Figure 6.1.

Figure 6.1 illustrates the number of vocabularies based on the number of the characters in the hyphenated corpus data in scenarios 1. Syllables with 2 and 3 characters are dominant. In scenario 2, the syllables have 6 characters as maximum size. The split vocab contain 10k frequently full-words including 5k full-words have number of characters from 2 up to 6.

To improve the vocabulary we add the most occurring full-words from the corpus data. We combine them with the top LNk (letter-N-grams) to create the language model and dictionary vocabulary.

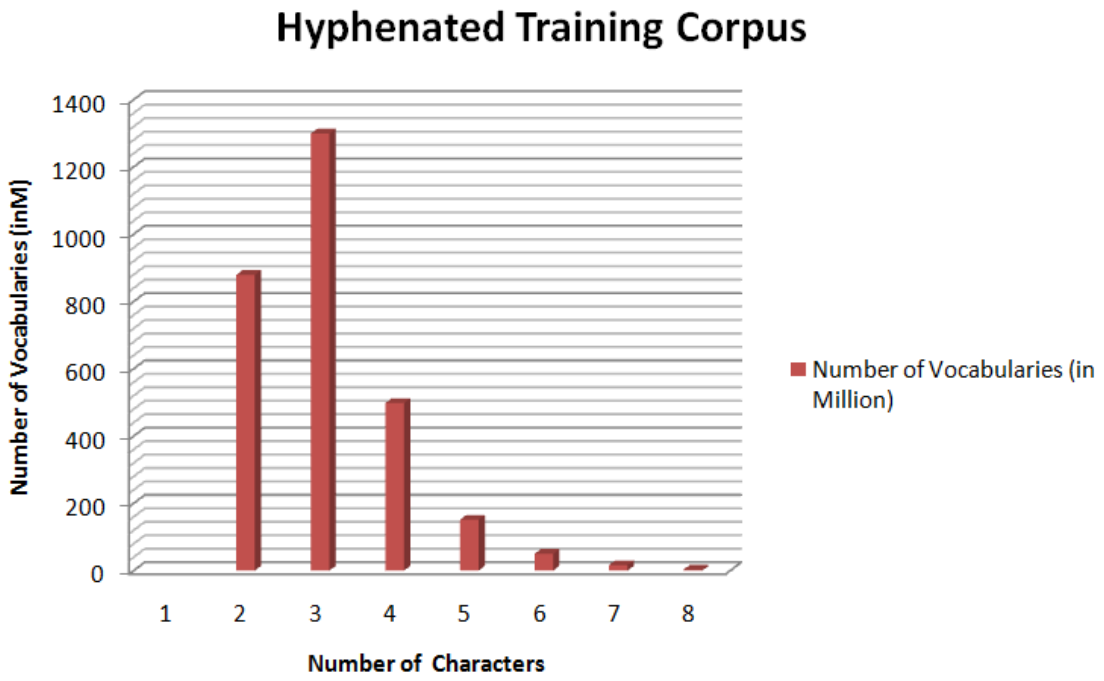

Figure 6.1: Number of Vocabularies based on the Number of Characters.

6.2.1 First Approach

In the first approach of scenario 2, we have two experiments. The first one based on the original created language model and dictionary vocabulary. In the second, attaching a "+" sign to the end of all language model and dictionary vocabulary.

6.2.1.1 First Experiment of 1.App of Scenario 2

As described in Figure 6.2, there is improvement in the Result of WER if we increase the size of the sub-words and fix the size of the full-words. Here we have increasing at the same time the size of the letter n-gram and the size of the total vocabulary. Therefore we can not summarize anything.

To find out which parameter has the influence on the result we have combine the top Wk full-word and the top LNk letter n-gram so that the total size of the vocabulary is maximal 100k. We varied LNk between 5k and 95k and Wk between 95k and 5k as shown in table 6.4.

In some experiments of this first approach, we tried the Morfessor tool as a supplementary method to split the training corps. Which did not deliver better results and was unstable in many experiments.

The WER and the perplexity are proportional to the augmentation of the percentage of the full-words in the 100k vocabularies. And this tabl 6.4 shows that we have no gain if we add the letter n-grams to the full-words to create the language model and dictionary vocabulary.

Figure 6.2: Improvement in the Result of WER by fixing the Size of the Full-Words and varying the Size of the Sub-Words.

LM	Full-words	Letter n-gram	Full-Compound Split		Morfessor	
			WER [%]	PPL	WER [%]	PPL
4-gram	5k	95k	40.1	199	-	733
	10k	90k	35.7	218	-	766
	20k	80k	32.9	234	51.3	769
	30k	70k	31.4	245	52.6	769
	40k	60k	30.2	252	-	769
	50k	50k	30.1	258	51.5	769
	60k	40k	29.1	263	-	769
	70k	30k	28.9	267	50.3	768
	100k	-	**28.6**	276	-	-

Table 6.4: Experimental Results of experiment 1 of Scenario 2.

6.2.1.2 Second Experiment of 1.App of Scenario 2

Add a sign "+" at the end of all vocabularies that means that the new vocabulary size is 200k distributed as follows: 100k vocabularies contain original full-words and letters n-gram and 100k vocabularies with suffix "+" at the end. This modification of the vocabulary size and adding a suffix like "+" to the half of the vocabularies help us to have the results presented in table 6.5. The minimum observed WER is achieved using 50k letters n-gram and 50k full-words. A clearer description 100k (50k with suffix "+" + 50k) letters n-gram and 100k (50k with suffix "+" + 50k) full-words. These result can prove that using big vocabulary size improve the WER in this thesis.

LM	Full-words	Letter n-gram	Full-Compound Split Script	
			WER [%]	PPL
4-gram	10k	90k	32.7	207
	20k	80k	30.3	221
	30k	70k	29.1	231
	50k	50k	**27.3**	242

Table 6.5: Experimental Results of Experiment 2 of Scenario 2 based on Adding the Suffix "+" to 100k vocabularies.

6.2.2 Second Approach

In the second approach we use the most frequent 10k letter n-grams to split the training corpus. The split is doing through the "Full-Compound Split Script". Top Wk and top LNk together represent the needed vocabulary to build the dictionary and the language model. The table 6.6 shows that increasing the full-words size from 50k to 95k and decreasing the letter n-grams size from 50k to 5k improve 1,2% the WER. This approach is characterized by the higher perplexities results.

LM	Full-words	Letter n-gram	Full-Compound Split Script	
			WER [%]	PPL
4-gram	50k	50k	36.5	1231
	60k	40k	35.9	1249
	70k	30k	35.7	1265
	80k	20k	35.6	1315
	90k	10k	35.6	1293
	95k	5k	35.3	938

Table 6.6: Experimental Results of 2 Approach of Scenario 2.

6.2.3 Third Approach

In the third approach both the Top 10k Wk and top 10k LNk represent the new split vocabulary with new size 20k. This approach is characterized with the rapprochement between the different WER results. Only 0.6% difference in the WER between the first cell of the table 6.7 and the last one. 29.7% is the lowest WER result.

LM	Full-words	Letter n-gram	Full-Compound Split Script	
			WER [%]	PPL
4-gram	50k	50k	30.9	345
	60k	40k	30.3	356
	70k	30k	30.3	359
	80k	20k	30	361
	90k	10k	29.8	363
	95k	5k	**29.7**	336

Table 6.7: Experimental Results of 3 Approach of Scenario 2.

Based on this approach we decide to use for the third scenario a combination of full-words and syllables to split the training corpus.

6.3 Experiments of Scenario 3

In scenario 3 the training corpus is sorted and the most frequent full-words Wk is taken. After the hyphenation of the sorted training corpus, the hyphenated corpus will be sorted and the most frequent syllables Sk is taken. In the scenario 3 adding the top Wk to the split vocabulary. Adding the vocabulary of characters to the top Wk and top Sk to build the language model and the dictionary it is what distinguishes the scenario 3.

To find the best combination of full-words and Syllables to have the best result, table 6.8 introduces a set of experiments in which the size of full-word is increased gradually up to 95k. And the size of syllables is decreased from 95k up to 5k.

LM	Full-words	Syllables	Full-Compound Split	
			WER [%]	PPL
4-gram	1k	30k	38.1	60
	5k	5k	32.8	69
	5k	15k	32.3	91
	5k	95k	32	98
	10k	90k	30	125
	20k	5k	29.1	124
	20k	30k	28.3	154
	20k	80k	28.2	158
	30k	70k	27.7	180
	40k	5k	28	161
	40k	30k	27.4	191
	40k	60k	27.4	196
	50k	50k	26.9	207
	60k	40k	26.4	215
	70k	30k	26.5	221
	80k	20k	**26.3**	225
	90k	10k	**26.3**	223
	95k	5k	26.4	214

Table 6.8: Experimental Results of Scenario 3 (4-gram).

In tabel 6.8 the minimum observed WER is achieved in two combinations using 80k full-words + 20k syllables and 90k full-words and 10k syllables.

6.4 How to improve the WER in Scenario 3?

In this section we describe the 3 ideas doing to improve the WER. These are using 6-gram language model instead of 4-gram and using a grapheme dictionary. At the end adding syllables pronunciation variation and characters variations to the dictionary.

6.4.1 6-gram Language model

The table 6.9 shows a summary of the best achieved WERs based on a 6-gram language model. We observe here that we have almost the same results as 4-gram LM or 0,1% worse. But the results of the perplexities is better as the 4-gram LM.

LM	Full-words	Syllables	Full-Compound Split Script	
			WER [%]	PPL
6 n-gram	50k	50k	27	204
	60k	40k	26.5	212
	70k	30k	26.5	218
	80k	20k	**26.3**	221
	90k	10k	26.4	217
	95k	5k	26.4	207

Table 6.9: Experimental Results of Scenario 3 (6 n-gram).

6.4.2 Grapheme Dictionary

In addition to systems with a phoneme-based dictionary, we also built grapheme-based recognition system By using the same data and only replacing the original pronunciation dictionary with grapheme dictionary which is a 1:1 mapping approach between letters and sounds.

The table 6.10 shows that in this ASR system the using of the grapheme dictionary not improve the WER.

LM	Full-words	Syllables	Full-Compound Split Script
			WER [%]
4-gram	50k	50k	29
	60k	40k	28.1
	70k	30k	27.2
	80k	20k	27.3
	90k	10k	27.2
	95k	5k	27.4

Table 6.10: Experimental Results of Scenario 3 based on Grapheme Dictionary.

6.4.3 Syllables and Characters Pronunciation Variants

In this section we add another one syllables and from two up to three characters pronunciations variants to the original dictionary. Using the G2P sequitur to generate 10 pronunciations variants for every character. Then we select the best two or three pronunciations. After performing this approach the average reduction in the WER is around 0.3% compared to the use of one pronunciation variant for syllables. The tables 6.11 shows that the minimum observed WER, 26%, is achieved using 60k full-words and 40k syllables.

LM	Full-words	Syllables	Full-Compound Split Script
			WER [%]
4-gram	50k	50k	26.4
	60k	40k	**26**
	70k	30k	26.2
	80k	20k	26.1
	90k	10k	26.3
	95k	5k	26.2

Table 6.11: Experimental Results of Scenario 3 based on Syllables and Characters Pronunciation Variants.

6.5 Statistics of the best Results

In this section we give an overview about the split training corpus of the 6 best experiments results based on statistics. Then in detail for the ASR system with 26% WER result.

The table 6.12 represents the statistics Results for split training corpus for the best 6 Experiments WER Results. The first thing that can be seen is the high proportion of the full words in the split training data. We remember that the term word means the vocabulary in a sentence between two spaces. This means that the top Wk frequently full words contain the different prepositions, conjunctions and German articles. These are present almost in all German sentence. And the figure 6.3 shows the number of occurrences of some of them in the sorted training corpus.

```
11419224 für
11460669 auf
11550180 ein
11902215 im
12585951 zu
13212492 von
13406567 mit
16326796 das
16892660 den
26485307 in
42472161 und
48166573 der
49383113 die
```

Figure 6.3: The most frequently words in the training corpus data.

95.2% is the largest obtained proportion by 100k vocabularies contain 95k full words and 5k syllables. 40% from 95k full words are words contain up to 6 characters.

As it's known the quality of the split of the corpus data related to the given split vocabulary. Therefore, the explanation of the exist of the vocabulary split with character is the absence of the proper syllable in the split vocabulary to split it.

The percentage of the vocabulary split with character is a metric to evaluate the quality of the syllables. The lower percentage of the vocabulary split with character, the higher is the quality of the syllable. Table 6.12 shows the percentage of the full-words and the percentage of the vocabulary split into syllables and characters in the 6 different training corpus. As we see increasing the size of the full-word and decreasing the size of the syllable increase the size of the vocabulary split into characters. And we can understand that because of the need to make up the shortfall in the amount of syllable vocabulary.

Full-words	Syllables	Split training Data		
		Full-Words [%]	Syllables [%]	Vocab. split into Characters [%]
50k	50k	93.19	6.69	0.12
60k	40k	93.9	5.94	0.16
70k	30k	94.45	5.32	0.23
80k	20k	94.88	4.79	0.33
90k	10k	94.24	4.32	0.44
95k	5k	95.5	3.96	0.54

Table 6.12: Statistics Results for split Training Corpus for the best 6 Experiments WER Results.

The table 6.13 represents the results of the split 10 texts based on 60k full words and 40k syllable. Despite different sizes and the number of contained vocabulary the 10 split texts maintained almost the same quote of full words, syllables and vocabulary split with character.

Full-words	Syll.	Split Text	Full-Words [%]	Syll. [%]	Vocab. split into Characters [%]
60k	40k	1	96.02	3.94	0.04
		2	94.47	5.44	0.09
		3	93.29	6.7	0.01
		4	93.45	6.43	0.12
		5	93.37	6.51	0.12
		6	93.36	6.52	0.12
		7	93.31	6.55	0.14
		8	92.39	7.39	0.22
		9	93.15	6.69	0.16
		10	95.90	4.07	0.03

Table 6.13: Statistics Results for split 10 texts based on 60k Full-words and 40k Syllables.

The table 6.14 below shows some full-words output examples which represent our objective in this thesis

Nutzbarkeit (usability)	nutz+ bar+ keit
Spielzeug (toys)	spiel+ zeug
Diagnose (diagnosis)	dia+ gno+ se
Konzertsaal (concert Hall)	kon+ zert+ saal
Vollzusage (voll engagement)	voll+ zu+ sa+ ge
Überwachungsproblem (monitoring problem)	über+ wa+ chungs+ pro+ blem

Table 6.14: Some Examples of Compounding Sequences of Sub-Units to form Full-Words.

7. Summary and future work

In this thesis, we built a LVCSR system for the German language based on sub-words. The aim is to overcome the lack of vocabulary in the regular full-word speech recognition systems by generating new vocabulary from the training data, which occurs especially with morphological rich languages such as German. The decision of using a sub-words based system is due to their capability to model unseen words in the training corpus by compounding sequences of sub-units forming new words.

Our experiments are conducted on a German training corpus containing about 9 million full-words. In three scenarios, we investigated the results of different techniques acting on the input split, the language model and the dictionary vocabulary. Two techniques are involved in generating sub-words, which are syllables and letter n-grams.

Our experiments show that using a combination of full-words and syllables from the training corpus leads to the best WER results. Improvement gains come from adding the syllables and characters pronunciations variants to the dictionary, while no gain is achieved by replacing the dictionary with a grapheme dictionary.

The main drawback of the sub-word based approach is that the degree of acoustic confusion among different recognition units becomes higher, which is due to the short length of the units. Therefore, as future work we propose a more driven and restricted use of sub-words. For example, by fixing a maximum and a minimum length of units, or by finding out syllables that have a meaning. We also suggest investigating other sub-word extraction techniques, such as morpheme.

A different approach for estimating sub-words based Language Models could be the use of feed-forward deep neural networks (DNNs).

Finally, we find it interesting to try our approach on building ASR for other morphological rich languages, such as Arabic and Polish.

Bibliography

[ACP+09] E. Arisoy, D. Can, S. Parlak, H. Sak, and M. Saraçlar, "Turkish broadcast news transcription and retrieval," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 5, pp. 874–883, 2009.

[AD03] M. Adda-Decker, "A corpus-based decompounding algorithm for german lexical modeling in lvcsr," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[ADA00] M. Adda-Decker and G. Adda, "Morphological decomposition for asr in german," in *Workshop on Phonetics and Phonology in Automatic Speech Recognition*, 2000, pp. 129–143.

[ADdMAL05] M. Adda-Decker, P. B. de Mareüil, G. Adda, and L. Lamel, "Investigating syllabic structures and their variation in spontaneous french," *Speech Communication*, vol. 46, no. 2, pp. 119–139, 2005.

[ADL99] M. Adda-Decker and L. Lamel, "Pronunciation variants across system configuration, language and speaking style," *Speech Communication*, vol. 29, no. 2, pp. 83–98, 1999.

[Ass99] I. P. Association, *Handbook of the International Phonetic Association: A guide to the use of the International Phonetic Alphabet*. Cambridge University Press, 1999.

[AT08] M. M. Azmi and H. Tolba, "Syllable-based automatic arabic speech recognition in noisy enviroment," in *Audio, Language and Image Processing, 2008. ICALIP 2008. International Conference on*. IEEE, 2008, pp. 1436–1441.

[AV03] W. W. Anand Venkataraman, "Select-vocab — sri international," 2003. [Online]. Available: http://www.speech.sri.com/projects/srilm/manpages/select-vocab.1.html

[Bak76] R. Bakis, "Continuous speech recognition via centisecond acoustic states," *The Journal of the Acoustical Society of America*, vol. 59, no. S1, pp. S97–S97, 1976.

[BAY58] T. BAYES, "Studies in the history of probability and statistics: Ix thomas bayes's essay towards solving a problem in the doctrine of chances," *Biometrika*, vol. 45, no. 3/4, pp. 293–315, December 1958.

[BBV04] H. Bunke, S. Bengio, and A. Vinciarelli, "Offline recognition of unconstrained handwritten texts using hmms and statistical language models," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 26, no. 6, pp. 709–720, 2004.

[BdSG+91] L. R. Bahl, P. V. de Souza, P. Gopalakrishnan, D. Nahamoo, and M. Picheny, "Context dependent modeling of phones in continuous speech using decision trees." in *HLT*, 1991.

[BFRB96] A. Berton, P. Fetter, and P. Regel-Brietzmann, "Compound words in large-vocabulary german speech recognition systems," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 2. IEEE, 1996, pp. 1165–1168.

[BK05] S. Broman and M. Kurimo, "Methods for combining language models in speech recognition," 2005.

[BN03] M. Bisani and H. Ney, "Multigram-based grapheme-to-phoneme conversion for lvcsr." in *INTERSPEECH*, 2003.

[BN05] ——, "Open vocabulary speech recognition with flat hybrid models." in *INTERSPEECH*, 2005, pp. 725–728.

[BN08] ——, "Joint-sequence models for grapheme-to-phoneme conversion," *Speech Communication*, vol. 50, no. 5, pp. 434–451, 2008.

[C+06] M. Creutz *et al.*, *Induction of the morphology of natural language: Unsupervised morpheme segmentation with application to automatic speech recognition*. Helsinki University of Technology, 2006.

[CG96] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 1996, pp. 310–318.

[CHK+07] M. Creutz, T. Hirsimäki, M. Kurimo, A. Puurula, J. Pylkkönen, V. Siivola, M. Varjokallio, E. Arisoy, M. Saraçlar, and A. Stolcke, "Morph-based speech recognition and modeling of out-of-vocabulary words across languages," *ACM Transactions on Speech and Language Processing (TSLP)*, vol. 5, no. 1, p. 3, 2007.

[CL02] M. Creutz and K. Lagus, "Unsupervised discovery of morphemes," in *Proceedings of the ACL-02 workshop on Morphological and phonological learning-Volume 6*. Association for Computational Linguistics, 2002, pp. 21–30.

[CL05] ——, *Unsupervised morpheme segmentation and morphology induction from text corpora using Morfessor 1.0*. Helsinki University of Technology, 2005.

[CLV06] M. Creutz, K. Lagus, and S. Virpioja, "Unsupervised morphology induction using morfessor," in *Finite-State Methods and Natural Language Processing*. Springer, 2006, pp. 300–301.

[CPCZ06] G. Choueiter, D. Povey, S. F. Chen, and G. Zweig, "Morpheme-based language modeling for arabic lvcsr," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[Dar15a] Dartmouth, "Komposita — dartmouth," 2015. [Online]. Available: http://www.dartmouth.edu/~german/Grammatik/Wortbildung/Komposita.html

[Dar15b] ——, "Prefixes — dartmouth," 2015. [Online]. Available: http://www.dartmouth.edu/~german/Grammatik/Wortbildung/Separables.html

[Dar15c] ——, "Suffixes — dartmouth," 2015. [Online]. Available: http://www.dartmouth.edu/~german/Grammatik/Wortbildung/Suffixes.html

[Dij59] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[Dou98] S. C. Douglas, "Evaluation metrics for language models," 1998.

[EDGR+09] A. El-Desoky, C. Gollan, D. Rybach, R. Schlüter, and H. Ney, "Investigating the use of morphological decomposition and diacritization for improving arabic lvcsr." in *INTERSPEECH*, 2009, pp. 2679–2682.

[EDMSSN10] A. El-Desoky Mousa, M. A. B. Shaik, R. Schluter, and H. Ney, "Sub-lexical language models for german lvcsr," in *Spoken Language Technology Workshop (SLT), 2010 IEEE*. IEEE, 2010, pp. 171–176.

[Fea12] S. Feathersto, "Introduction to general linguistics." 2012. [Online]. Available: http://www.sfs.uni-tuebingen.de/~sam/teach/IntroGenLing/handouts/syn1HO.pdf

[FGH+97] M. Finke, P. Geutner, H. Hild, T. Kemp, K. Ries, and M. Westphal, "The karlsruhe-verbmobil speech recognition engine," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 83–86.

[Fis06] J. Fiscus. (2006) Sclite - score speech recognition system output. [Online]. Available: http://www.icsi.berkeley.edu/Speech/docs/sctk-1.2/sclite.htm

[FL99] J. E. Fosler-Lussier, "Dynamic pronunciation models for automatic speech recognition," Ph.D. dissertation, University of California, Berkeley Fall 1999., 1999.

[Gal03] L. Galescu, "Recognition of out-of-vocabulary words with sub-lexical language models," in *Eighth European Conference on Speech Communication and Technology*, 2003.

[GNW95] M. Generet, H. Ney, and F. Wessel, "Extensions of absolute discounting for language modeling," in *Fourth European Conference on Speech Communication and Technology*, 1995.

[GO15] R. Gutierrez-Osuna, "Introduction to speech processing." 2015. [Online]. Available: http://research.cs.tamu.edu/prism/lectures/sp/l15.pdf

[HCS+06] T. Hirsimäki, M. Creutz, V. Siivola, M. Kurimo, S. Virpioja, and J. Pylkkönen, "Unlimited vocabulary speech recognition with morph language models applied to finnish," *Computer Speech & Language*, vol. 20, no. 4, pp. 515–541, 2006.

[HHSL05] T. J. Hazen, I. L. Hetherington, H. Shu, and K. Livescu, "Pronunciation modeling using a finite-state transducer representation," *Speech Communication*, vol. 46, no. 2, pp. 189–203, 2005.

[HM14] T. H. Hlaing and Y. Mikami, "Automatic syllable segmentation of myanmar texts using finite state transducer," *ICTer*, vol. 6, no. 2, 2014.

[Hol01] J. Holmes, *An introduction to sociolinguistics*, 2nd ed. Harlow, Eng. ; New York : Longman, 2001.

[HPK09] T. Hirsimaki, J. Pylkkonen, and M. Kurimo, "Importance of high-order n-gram models in morph-based speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 17, no. 4, pp. 724–732, 2009.

[ID10] N. Indurkhya and F. J. Damerau, *Handbook of natural language processing*. CRC Press, 2010, vol. 2.

[Jel69] F. Jelinek, "Fast sequential decoding algorithm using a stack," *IBM Journal of Research and Development*, vol. 13, no. 6, pp. 675–685, 1969.

[JWB+01] D. Jurafsky, W. Ward, Z. Banping, K. Herold, Y. Xiuyang, and Z. Sen, "What kind of pronunciation variation is hard for triphones to model?" in *Acoustics, Speech, and Signal Processing, 2001. Proceedings.(ICASSP'01). 2001 IEEE International Conference on*, vol. 1. IEEE, 2001, pp. 577–580.

[k1215] k12reader, "Compound-words — k12reader," 2015. [Online]. Available: http://www.k12reader.com/term/compound-words/

[Kat87] S. Katz, "Estimation of probabilities from sparse data for the language model component of a speech recognizer," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 35, no. 3, pp. 400–401, 1987.

[KHM+] K. Kilgour, M. Heck, M. Müller, M. Sperber, S. Stüker, and A. Waibel, "The 2014 kit iwslt speech-to-text systems for english, german and italian."

[Kil09] K. Kilgour, "Language model adaptation using interlinked semantic data," 2009.

[KJ96] T. Kemp and A. Jusek, "Modelling unknown words in spontaneous speech," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 1. IEEE, 1996, pp. 530–533.

[KK01] J. Kneissler and D. Klakow, "Speech recognition for huge vocabularies by using optimized sub-word units." in *INTERSPEECH*, 2001, pp. 69–72.

[KN95] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 1. IEEE, 1995, pp. 181–184.

[KP02] D. Klakow and J. Peters, "Testing the correlation of word error rate and perplexity," *Speech Communication*, vol. 38, no. 1, pp. 19–28, 2002.

[Lee96] P. Lee, *The Whorf theory complex: a critical reconstruction*. John Benjamins Publishing, 1996, vol. 81.

[Lev66] V. I. Levenshtein, "Binary codes capable of correcting deletions, insertions, and reversals," in *Soviet physics doklady*, vol. 10, no. 8, 1966, pp. 707–710.

[LFLM12] K. Livescu, E. Fosler-Lussier, and F. Metze, "Subword modeling for automatic speech recognition: Past, present, and emerging approaches," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 44–57, 2012.

[Lia83] F. M. Liang, *Word hyphenation by computer*. Department of Computer Science, Stanford University, 1983.

[Lid20] G. J. Lidstone, "Note on the general case of the bayes-laplace formula for inductive or a posteriori probabilities," *Transactions of the Faculty of Actuaries*, vol. 8, no. 182-192, p. 13, 1920.

[LJ14] P. Ladefoged and K. Johnson, *A course in phonetics*. Cengage learning, 2014.

[LJSR89] C.-H. Lee, B.-H. Juang, F. K. Soong, and L. Rabiner, "Word recognition using whole word and subword models," in *Acoustics, Speech, and Signal Processing, 1989. ICASSP-89., 1989 International Conference on*. IEEE, 1989, pp. 683–686.

[LPR+03] Y.-S. Lee, K. Papineni, S. Roukos, O. Emam, and H. Hassan, "Language model based arabic word segmentation," in *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*. Association for Computational Linguistics, 2003, pp. 399–406.

[LSF14] M. P. Lewis, G. F. Simons, and C. D. Fennig, *Ethnologue: Languages of the World. 17th edition.* SIL International, 2014.

[LWKR00] M. Larson, D. Willett, J. Köhler, and G. Rigoll, "Compound splitting and lexical unit recombination for improved performance of a speech recognition system for german parliamentary speeches." in *INTERSPEECH*, 2000, pp. 945–948.

[LWL+97] A. Lavie, A. Waibel, L. Levin, M. Finke, D. Gates, M. Gavalda, T. Zeppenfeld, and P. Zhan, "Janus-iii: Speech-to-speech translation in multiple languages," in *Acoustics, Speech, and Signal Processing, 1997. ICASSP-97., 1997 IEEE International Conference on*, vol. 1. IEEE, 1997, pp. 99–102.

[Maj08] P. Majewski, "Syllable based language model for large vocabulary continuous speech recognition of polish," in *Text, Speech and Dialogue*. Springer, 2008, pp. 397–401.

[MGSN98] D. McAllaster, L. Gillick, F. Scattone, and M. Newman, "Fabricating conversational speech data with acoustic models: a program to examine model-data mismatch." in *ICSLP*, 1998.

[Mih11] V. Mihalicek, *Language Files.* Ohio State University, 2011.

[Mou14] A. I. E.-D. Mousa, "Sub-word based language modeling of morphologically rich languages for lvcsr," Ph.D. dissertation, Universitätsbibliothek, 2014.

[MR+02] S. J. Melnikoff, M. J. Russell *et al.*, "Speech recognition on an fpga using discrete and continuous hidden markov models," in *Field-Programmable Logic and Applications: Reconfigurable Computing Is Going Mainstream*. Springer, 2002, pp. 202–211.

[MSD+12] T. Mikolov, I. Sutskever, A. Deoras, H.-S. Le, S. Kombrink, and J. Cernocky, "Subword language modeling with neural networks," *preprint (http://www. fit. vutbr. cz/imikolov/rnnlm/char. pdf)*, 2012.

[MSSN13] A. E.-D. Mousa, M. A. B. Shaik, R. Schlüter, and H. Ney, "Morpheme level hierarchical pitman-yor class-based language models for lvcsr of morphologically rich languages." in *INTERSPEECH*. Citeseer, 2013, pp. 3409–3413.

[NEK94] H. Ney, U. Essen, and R. Kneser, "On structuring probabilistic dependences in stochastic language modelling," *Computer Speech & Language*, vol. 8, no. 1, pp. 1–38, 1994.

[Ném06] L. Németh, "Automatic non-standard hyphenation in openoffice. org," *TUGboat*, vol. 27, no. 1, pp. 32–37, 2006.

[NHUTO92] H. Ney, R. Haeb-Umbach, B.-H. Tran, and M. Oerder, "Improvements in beam search for 10000-word continuous speech recognition," in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 1. IEEE, 1992, pp. 9–12.

[NMW97] H. Ney, S. Martin, and F. Wessel, "Statistical language modeling using leaving-one-out," in *Corpus-based methods in Language and Speech processing*. Springer, 1997, pp. 174–207.

[OSS05] M. Ostendorf, E. Shriberg, and A. Stolcke, "Human language technology: Opportunities and challenges," DTIC Document, Tech. Rep., 2005.

[OVHDJ03] R. Ordelman, A. Van Hessen, and F. De Jong, "Compound decomposition in dutch large vocabulary speech recognition." in *INTERSPEECH*, 2003.

[Par11] M. C. Parada, *Learning sub-word units and exploiting contextual information for open vocabulary speech recognition*. Citeseer, 2011.

[Pau91] D. B. Paul, "Algorithms for an optimal a* search and linearizing the search in the stack decoder," in *Acoustics, Speech, and Signal Processing, 1991. ICASSP-91., 1991 International Conference on*. IEEE, 1991, pp. 693–696.

[Rab89] L. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[Rei05] K. K.-S. Reimann, "Basiswissen deutsche gegenwartssprache," 2005.

[RJ86] L. Rabiner and B.-H. Juang, "An introduction to hidden markov models," *ASSP Magazine, IEEE*, vol. 3, no. 1, pp. 4–16, 1986.

[RW95] I. Rogina and A. Waibel, "The janus speech recognizer," in *ARPA SLT Workshop*, 1995, pp. 166–169.

[S+02] A. Stolcke *et al.*, "Srilm-an extensible language modeling toolkit." in *INTERSPEECH*, 2002.

[Sch08] T. Schultz. (Sommersemester 2008) Kit vorlesung multilinguale mensch-maschine kommunikation. [Online]. Available: http://csl.ira.uka.de/fileadmin/Vorlesungen/SS2008/MMMK

[Sch12] ——. (Sommersemester 2012) Kit vorlesung multilinguale mensch-maschine kommunikation. [Online]. Available: http://csl.anthropomatik.kit.edu/downloads/vorlesungsinhalte/MMMK-PP14-AcousticModeling2-SS2012.pdf

[Sch14] T. Schlippe, "Rapid generation of pronunciation dictionaries for new domains and languages," Ph.D. dissertation, Karlsruhe, Karlsruher Institut für Technologie (KIT), Diss., 2014, 2014.

[SDSV] I. T. Schultz, E. G. K. Djomgang, D.-I. T. Schlippe, and D.-I. T. Vu, "Hausa large vocabulary continuous speech recognition."

[SK06] T. Schultz and K. Kirchhoff, *Multilingual speech processing*. Academic Press, 2006.

[SLE05] C. Schrumpf, M. Larson, and S. Eickeler, "Syllable-based language models in speech recognition for english spoken document retrieval," in *Proceedings of the 7th International Workshop of the EU Network of excellence DELOS on Audio-Visual Content and Information Visualization in Digital Libraries*, 2005, pp. 196–205.

[SMFW01] H. Soltau, F. Metze, C. Fugen, and A. Waibel, "A one-pass decoder based on polymorphic linguistic context assignment," in *Automatic Speech Recognition and Understanding, 2001. ASRU'01. IEEE Workshop on*. IEEE, 2001, pp. 214–217.

[SMSN11a] M. A. B. Shaik, A.-D. Mousa, R. Schluter, and H. Ney, "Using morpheme and syllable based sub-words for polish lvcsr," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4680–4683.

[SMSN11b] M. A. B. Shaik, A. E.-D. Mousa, R. Schlüter, and H. Ney, "Hybrid language models using mixed types of sub-lexical units for open vocabulary german lvcsr." in *INTERSPEECH*, 2011, pp. 1441–1444.

[SN02] A. Sixtus and H. Ney, "Training of across-word phoneme models for large vocabulary continuous speech recognition," in *IEEE INTERNATIONAL CONFERENCE ON ACOUSTICS SPEECH AND SIGNAL PROCESSING*, vol. 1. Citeseer, 2002, pp. I–849.

[SOA09] T. Shinozaki, M. Ostendorf, and L. Atlas, "Characteristics of speaking style and implications for speech recognition," *The Journal of the Acoustical Society of America*, vol. 126, no. 3, pp. 1500–1510, 2009.

[SSG10] H. Sak, M. Saraclar, and T. Gungor, "Morphology-based and sub-word language modeling for turkish speech recognition," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5402–5405.

[StÃ12] S. StÃ$\frac{1}{4}$ker. (2012) Kit - janus recognition toolkit. [Online]. Available: http://isl.anthropomatik.kit.edu/english/1406.php

[SVG+14] P. Smit, S. Virpioja, S.-A. Grönroos, M. Kurimo *et al.*, "Morfessor 2.0: Toolkit for statistical morphological segmentation," in *The 14th Conference of the European Chapter of the Association for Computational Lin-*

guistics (EACL), Gothenburg, Sweden, April 26-30, 2014. Aalto University, 2014.

[Tra59] G. L. Trager, "The systematization of the whorf hypothesis," *Anthropological linguistics*, pp. 31–35, 1959.

[T.S88] E.-P. T.Svendsen, K.K.Paliwal, "Experiments with a sub word based speech recognizer," *preprint (http://www.assta.org/sst/SST-88/cache/SST-88-Chapter9-p14.pdf)*, 1988.

[VKV13] M. Varjokallio, M. Kurimo, and S. Virpioja, "Learning a subword vocabulary based on unigram likelihood." in *ASRU*, 2013, pp. 7–12.

[VVCS07] S. Virpioja, J. J. Väyrynen, M. Creutz, and M. Sadeniemi, "Morphology-aware statistical machine translation based on morphs induced in an unsupervised manner," *Machine Translation Summit XI*, vol. 2007, pp. 491–498, 2007.

[W+97] J. C. Wells *et al.*, "Sampa computer readable phonetic alphabet," *Handbook of standards and resources for spoken language systems*, vol. 4, 1997.

[w3t15] w3techs, "Usage statistics of content languages for websites, january 2015," 2015. [Online]. Available: http://w3techs.com/technologies/overview/content_language/all

[Wik15a] Wikipedia, "German language — Wikipedia, the free encyclopedia," 2015. [Online]. Available: http://en.wikipedia.org/wiki/German_language#cite_note-12

[Wik15b] ——, "Root (linguistics) — Wikipedia, the free encyclopedia," 2015. [Online]. Available: http://en.wikipedia.org/wiki/Root_(linguistics)

[WTHSS96] M. Weintraub, K. Taussig, K. Hunicke-Smith, and A. Snodgrass, "Effect of speaking style on lvcsr performance," in *Proc. ICSLP*, vol. 96. Citeseer, 1996, pp. 16–19.

[XMZ+96] B. Xu, B. Ma, S. Zhang, F. Qu, and T. Huang, "Speaker-independent dictation of chinese speech with 32k vocabulary," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 4. IEEE, 1996, pp. 2320–2323.

[XNN+06] B. Xiang, K. Nguyen, L. Nguyen, R. Schwartz, and J. Makhoul, "Morphological decomposition for arabic broadcast news transcription," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[You96] S. Young, "A review of large-vocabulary continuous-speech," *Signal Processing Magazine, IEEE*, vol. 13, no. 5, p. 45, 1996.

[YOW94] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proceedings of the workshop on Human Language Technology*. Association for Computational Linguistics, 1994, pp. 307–312.