AALTO UNIVERSITY

School of Electrical Engineering
Department of Signal Processing and Acoustics

Silja Tirronen

Automated Testing of Speech-to-Speech Machine Translation in Telecom Networks

Master's Thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Technology.

Espoo, February 21, 2011

Supervisor: Professor Paavo Alku
Instructor: Peter Jungner, PhD

Aalto University
School of Electrical Engineering

AALTO UNIVERISTY
SCHOOL OF ELECTRICAL ENGINEERING Abstract of the Master's Thesis

Author: Silja Tirronen

Name of the thesis: Automated Testing of Speech-to-Speech Machine Translation in Telecom Networks

Number of pages: 64

Date: February 21, 2011

Department: Department of Signal Processing and Acoustics

Professorship: S-89 Acoustics and Audio Signal Processing

Supervisor: Professor Paavo Alku

Instructor: Peter Jungner, PhD

In the globalizing world, the ability to communicate over language barriers is increasingly important. Learning languages is laborious, which is why there is a strong desire to develop automatic machine translation applications. Ericsson has developed a speech-to-speech translation prototype called the Real-Time Interpretation System (RTIS). The service runs in a mobile network and translates travel phrases between two languages in speech format.

The state-of-the-art machine translation systems suffer from a relatively poor performance and therefore evaluation plays a big role in machine translation development. The purpose of evaluation is to ensure the system preserves the translational equivalence, and in case of a speech-to-speech system, the speech quality. The evaluation is most reliably done by human judges. However, human-conducted evaluation is costly and subjective.

In this thesis, a test environment for Ericsson Real-Time Interpretation System prototype is designed and analyzed. The goals are to investigate if the RTIS verification can be conducted automatically, and if the test environment can truthfully measure the end-to-end performance of the system.

The results conclude that methods used in end-to-end speech quality verification in mobile networks can not be optimally adapted for machine translation evaluation. With current knowledge, human-conducted evaluation is the only method that can truthfully measure translational equivalence and the speech intelligibility. Automating machine translation evaluation needs further research, until which human-conducted evaluation should remain the preferred method in RTIS verification.

Keywords: natural language processing, automatic speech recognition, speech synthesis, machine translation, translational equivalence, user-plane verification, speech quality

AALTO-YLIOPISTO
SÄHKÖTEKNIIKAN KORKEAKOULU Diplomityön tiivistelmä

Tekijä: Silja Tirronen

Työn nimi: Puhemuotoisten konekäännösten automatisoitu testaus tietoliikenneverkoissa

Sivumäärä: 64

Päivämäärä: 21.2.2011

Laitos: Signaalinkäsittelyn ja akustiikan laitos

Professuuri: S-89 Akustiikka ja äänenkäsittelytekniikka

Valvoja: Professori Paavo Alku

Ohjaaja: FT Peter Jungner

Globalisoituvassa maailmassa kyky kommunikoida kielimuurien yli käy yhä tärkeämmäksi. Kielten opiskelu on työlästä ja siksi halutaan kehittää automaattisia konekäännösjärjestelmiä. Ericsson on kehittänyt prototyypin nimeltä Real-Time Interpretation System (RTIS), joka toimii mobiiliverkossa ja kääntää matkailuun liittyviä fraaseja puhemuodossa kahden kielen välillä.

Nykyisten konekäännösjärjestelmien suorituskyky on suhteellisen huono ja siksi testauksella on suuri merkitys järjestelmien suunnittelussa. Testauksen tarkoituksena on varmistaa, että järjestelmä säilyttää käännösekvivalenssin sekä puhekäännösjärjestelmän tapauksessa myös riittävän puheenlaadun. Luotettavimmin testaus voidaan suorittaa ihmisten antamiin arviointeihin perustuen, mutta tällaisen testauksen kustannukset ovat suuria ja tulokset subjektiivisia.

Tässä työssä suunniteltiin ja analysoitiin automatisoitu testiympäristö Real-Time Interpretation System –käännösprototyypille. Tavoitteina oli tutkia, voidaanko testaus suorittaa automatisoidusti ja pystytäänkö todellinen, käyttäjän havaitsema käännösten laatu mittaamaan automatisoidun testauksen keinoin.

Tulokset osoittavat että mobiiliverkoissa puheenlaadun testaukseen käytetyt menetelmät eivät ole optimaalisesti sovellettavissa konekäännösten testaukseen. Nykytuntemuksen mukaan ihmisten suorittama arviointi on ainoa luotettava tapa mitata käännösekvivalenssia ja puheen ymmärrettävyyttä. Konekäännösten testauksen automatisointi vaatii lisää tutkimusta, jota ennen subjektiivinen arviointi tulisi säilyttää ensisijaisena testausmenetelmänä RTIS-testauksessa.

Avainsanat: luonnollisten kielten prosessointi, automaattinen puheentunnistus, puhesynteesi, konekäännös, käännösekvivalenssi, käyttäjätason testaus, puheenlaatu

Acknowledgements

This Master’s thesis was written at Oy LM Ericsson Ab between August 2010 and February 2011. During the process I received help and guidance from several talented individuals.

I want to thank my supervisor, professor Paavo Alku for the encouraging comments and advice. I also want to thank my inspiring instructor at Ericsson, Peter Jungner for his valuable ideas and assistance during this process. Special compliments go to Tony Söderudd, Tatu Järvi, and David Sjöstrand for technical support.

This thesis would never have been written without the original innovator, Dietmar Fiedler and his expertise. I also want to thank Ericsson for supporting this innovation despite all the challenges.

Special thanks go to all my friends. It’s been a real pleasure to spend my student years with you. The fun we had during this time kept me relaxed and motivated, and I appreciate it more than you can imagine.

Finally, I want to give my special thanks to Nick, not only for the proofreading, but also for the love, support, and inspiration during this time.

Espoo, February 21, 2011

Silja Tirronen

Contents

List of abbreviations

AMR	Adaptive Multi-Rate
AMR-WB	Adaptive Multi-Rate WideBand
AN	Access Network
ASR	Automatic Speech Recognition
ATM	Asynchronous Transfer Mode
BLEU	BiLingual Evaluation Understudy
BP	Board Processor
BS	Base Sation
BSC	Base Station Controller
BTS	Base Transceiver Station
CFG	Context-Free Grammar
CN	Core Network
CPP	Connectivity Packet Platform
DA	Domain Action
DSP	Digital Signal Processor
DTMF	Dual-Tone Multi-Frequency
DUT	Device Under Test
ET	Exchange Terminal
FLS	Finite-State Language
FST	Finite-State Transducer
GERAN	GSM Edge Radio Access Network
GPB	General Purpose Board
GSM	Global System for Mobile communication
HMM	Hidden Markov Model
IER	Information Error Rate
IP	Internet Protocol
KBMT	Knowledge-Based Machine Translation
LTS	Letter-To-Sound
MEM	Maximum Entropy Model
METEOR	Metric for Evaluation of Translation with Explicit Ordering
MGW	Media Gateway
M-MGw	Ericsson Mobile Media Gateway
MOS	Mean Opinion Square
MPD	Media Processing Device
MSB	Media Stream Board
MSC	Mobile Switching Center
MT	Machine Translation
PCM	Pulse Code Modulation
PESQ	Perceptual Evaluation of Speech Quality
PSTN	Public Switched Telephone Network
RNC	Radio Network Controller
RTIS IWD	Real-Time Interpretation System InterWorking Device
RTIS	Real-Time Interpretation System

SER	Sentence Error Rate
SIM	Subscriber Identity Module
SL	Source Language
SMT	Statistical Machine Translation
TDM	Time Division Multiplexing
TGD	Traffic Generator Device
TL	Target Language
TSC	Transit Switching Center
TTCN3	Testing and Test Control Notation, version 3
TTS	Text-To-Speech
UE	User Equipment
UMTS	Universal Mobile Telecommunications System
UPLOAD-BE	User Plane Load generator – Bit Exact
UTRAN	Universal Terrestrial Radio Access Network
VMGW	Virtual Media GateWay
WER	Word Error Rate
XML	eXtensible Markup Language

1 Introduction

1.1 Motivation

As the world is becoming global and the possibilities of instant communication around the world are growing, the need to study foreign languages has also become increasingly important. Learning languages is not easy: even after a few years of studies people face problems when communicating in a foreign language. In the Hitchhiker's Guide to the Galaxy by Douglas Adams, a little Babel Fish would simultaneously translate any language in the universe to another when placed in one's ear. Having a Babel Fish would remove language barriers as well as the trouble of studying foreign languages, and allow us to communicate with anyone regardless of the person's background.

Speech is the most natural way for human beings to communicate. Speech can deliver information conveniently to one or more listeners in real time and carry a lot of nonverbal information on the side. Therefore, there's a strong motivation to develop high-quality, real-time speech-to-speech machine translation applications. Companies also have a growing interest in commercializing speech-to-speech machine translation applications. The business potential of integrated network services capable of translating between at least the biggest languages, like English, Mandarin Chinese, Spanish, Russian or Arabic, is huge. Among other companies, Ericsson has noticed this business potential and has developed a translation service prototype. The prototype is capable of close-to real time speech-to-speech translations between English and Mandarin Chinese.

However, current machine translation applications, including Ericsson's prototype, still have a long way to go before they reach the Babel Fish's level of functionality. To successfully perform machine translation requires expertise in various fields of science: most importantly cognitive science, cortical processes of the human brain, acoustics, speech technology and linguistics. The lack of powerful and accurate methods for research and measurement of brain functions is the main reason why state-of-the-art machine translation applications suffer from relatively poor performance. Even without setting any fluency requirements, machine translation applications can not be considered reliable as they very often fail in their fundamental task: conveying the source language meaning. Mistranslations can either turn the dialog into complete nonsense, or change the meanings so that the outcome is offensive, amusing, or just plain confusing. It is obvious that this will result in mutual incomprehension and even destroy the whole discussion.

There is a lot of progress still to be made in machine translation development. This creates the need for efficient tools to measure the progress and accurately estimate the performance of a machine translation application. How can we verify the improvements? How can we verify that the translation service has met a certain performance level before deployment? Considering the complex nature of the translation problem itself, answering these questions is not trivial.

1.2 Problem description

Several scientists [Sta10, Tom03, Alb09, Lop08] agree to the current status of machine translation technology: significant progress has been made during recent years, but machine translation is still extremely unreliable. Moreover, cross-cultural communication is prone to misunderstandings even when using skilled human translators. The consequences of the mistranslations and misunderstandings vary in severity, but in worst case scenario serious conflicts can arise.

This brings up the importance of verification in machine translation development. Like in any software development process, verification is an integral and important part. Speech-to-speech machine translation systems are particularly error-prone and the consequences of a false acceptance can have a huge impact on the system performance. Hence, verification plays a big role in machine translation development and needs to be conducted thoroughly. However, as Tomás et. al. [Tom03] state, no accurate, consistent and convenient machine translation evaluation method yet exists.

We can start by examining the problem from speech quality verification standpoint. It is another speech-to-speech process of which there is a lot of expert knowledge available. Ericsson as the world's leading mobile network manufacturer has very sophisticated test environment for speech quality verification. Fully automated test setup with specialized tools enable testers to run high volume tests employing bit exact analysis. However, this same automated test environment can not be directly applied in speech-to-speech translation verification, as it differs significantly from speech quality verification. In case of no translation service is applied, the network's capability of accurately reproducing the same signal in the receiver's end is being measured. In case of translation, the signal will be run through a very complex cognitive process with multiple correct outputs. The complex nature of the translation process brings about new challenges in verification: how to define the reference patterns? How to define a metric that can calculate if the machine translation matches any of the references, and which one is the closest match? How to define the acceptance level for a translation, and thus, when is the service good enough to go live?

Another key issue in machine translation verification is automation. Setting up human-conducted verification is initially simple: human brain possesses a strong language model and knowledge of the world, which are needed for successful evaluation. Despite the low initial effort, human-conducted verification is expensive in terms of resources and time. It is also subjective and not accurately reproducible. Therefore we need to design a test environment which can perform evaluation with little or no human intervention. Automated tests could be used as a complementary method to manual verification to speed up regression testing and enable system tests with high traffic load. The key challenge in designing automated test environment is to ensure that the test results correspond to human judgement, i.e. the automated test gives a truthful representation of machine translation quality.

1.3 Scope

The purpose of this thesis is to examine the key principles behind successful machine translation and machine translation verification. Based on the theoretical aspects, the possibility of automating machine translation verification in telecom networks will be investigated. An automated test method and a test setup for Ericsson's speech-to-speech real-time interpretation system (RTIS) service prototype is designed and analyzed. The key goals of this thesis can be summarized as follows:

- To examine how machine translation works and present the main methods applied in machine translation evaluation
- To examine what kind of methods can be used to verify the effects of speech quality on machine translation quality and comprehensibility
- To design an automated test environment for Ericsson's speech-to-speech machine translation service
- To evaluate the designed test environment, i.e. estimate if the test setup can adequately measure the actual quality and comprehensibility of the translations

The task of the test environment designed in this thesis is to measure the end-to-end performance of an integrated network RTIS service. This means that not only the actual translation algorithm is tested, but also possible problems arising from speech recognition, speech synthesis, transcoding, network disturbances or delay need to be verified. The design is based on Ericsson's existing test environment for speech quality testing and related tools. The goal is to examine if the tools, methods, and test setup used in speech quality testing can be adapted and applied in machine translation verification.

This thesis is structured as follows. Chapter 2 presents the background theory considering speech production and natural languages. Chapter 3 presents the principles of automatic speech recognition (ASR), and speech synthesis, or text-to-speech (TTS). These speech processing methods are needed in all systems involving a vocal interface between humans and machines, hence, in speech-to-speech machine translation systems they are required as well. Chapter 4 reviews the history and current state of machine translation development, as well as the main design principles of machine translation systems. In chapter 5, subjective and quantitative methods for machine translation evaluation are presented. Chapter 6 gives an overview to the Ericsson's network and presents the objective and functionality of RTIS prototype. Chapter 7 presents the automated test environment for RTIS verification constructed in this thesis project. Chapter 8 presents the results of the tests, and analyzes the applicability of the test setup and its capability of truthfully measuring the translation quality. Chapter 9 concludes.

2 Background

2.1 Speech

Speech is probably the oldest information sharing method in the world. Speech and language are very efficient means of communication. Speech transmission over an electrical network has been possible now for more than 100 years, in one form or another. Still, research of speech coding and transmission is ongoing and new methods are actively being developed to achieve better speech quality. The new codecs introduced in 3G, such as AMR-WB or G.719, serve as an example. Speech controlled human-machine interaction is another, attractive research topic, growing in popularity in recent years. In this chapter, the basics of speech and speech production are presented.

2.1.1 Speech production

Speech is produced by letting pre-inhaled air flow through the speech production system and influencing the air stream by different positioning of the speech organs. Originating from lungs, the airflow passes through trachea and larynx, where two small muscle folds, called the vocal folds, are located. After the vocal folds the air passes through to the vocal tract, a muscular passageway consisting of oral and nasal cavities. The vocal tract leads from glottis (the opening between the vocal folds) all the way to the lips and nose, from which the speech signal radiates out. The vocal folds open and close rapidly to create excitation, a buzzing sound that is modified in the vocal tract to produce different phonemes. This can be understood as a simplified source-filter model as presented figure 2.1. An excitation signal is fed into a time-varying filtering system corresponding to the vocal tract, which creates resonances by emphasizing certain frequencies. [Ros02, Sha00, Jur08]

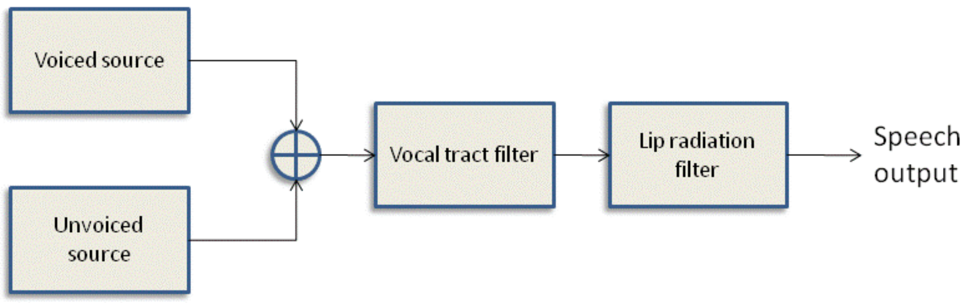

Figure 2.1. The source-filter model of speech.

The vocal tract is the most remarkable part of the speech production system. Its main function is to create acoustic resonances or restrict the airflow by changing its shape, and thus turn the excitation signals to specialized sounds of speech, also called phonemes. There are two main classes of phonemes: vowels and consonants. These two groups can be categorized further: consonants based on the place or manner of articulation, i.e. where and how in the vocal tract the airflow is restricted to produce the phoneme. Vowels are characterized mainly by the position of tongue and lips, and the openness of the vocal tract. [Jur08, Ros02]

Phonemes have their own characteristics but they are not fixed due to a phenomenon called coarticulation. That is, the articulation of a single phoneme may change remarkably depending on its phonetic context. The vocal tract tends to change its shape smoothly over a sequence of phonemes, which causes a single phoneme to be affected by the preceding and subsequent ones. [Sha00]

2.1.2 Speech signal properties

From a physical perspective, speech is variations in air pressure. The pressure changes propagate through air medium as sound waves. Speech signals can be characterized by their frequency, amplitude, waveform and spectral properties.

The speech production organs have evolved to work at relatively low frequencies, in the frequency range where also the hearing system is most sensitive. The fundamental frequency, commonly notated F_0, is perceived as the pitch of voice. The fundamental frequency is determined by the opening and closing of the vocal folds when producing voiced sounds. F_0 is speaker-dependent and varies within a wide frequency range. On average, male speakers' F_0 is approximately 100 Hz and female speakers' 200 Hz. The frequency range of human speech can go up to 10 kHz.

In the time-domain (or waveform) analysis, the speech signal is presented as a graph of air pressure variation as a function of time. In time domain representation, vowels can be observed to be quasi-periodic and have high amplitude and intensity. Their waveform resembles sine and cosine functions. The duration of a single vowel phoneme is relatively long, typically 50-400 ms. Vowels are voiced, so the opening and closing of the vocal folds can be clearly seen in the waveform. Consonants, however, are often noisy and don't have a regular waveform. Some resemble white noise, some are preceded by a period of silence and the duration of the actual sound burst is very short, some voiced consonant's waveforms can be rather similar to that of vowels. In general, consonants have shorter duration and lower intensity than vowels.

Frequency domain representations can give us more information about the characteristics of different phonemes. In frequency domain the signal's spectrum, i.e. the signal's amplitude as a function of frequency is presented. The spectral information is useful especially from speech coding and transmission point of view.

In frequency domain we can observe that most of the energy of a vowel lies below 1 kHz. Vowels also have an interesting spectral property known as formants. Depending on the shape of the vocal tract, some frequencies resonate more than others and thus spectral peaks appear. These peaks are called formants. All vowels have roughly three formants in the frequency range below 3 kHz. The locations of these first three formants are the most important features that make vowels distinguishable to the listener. Consonants' most important spectral information often lies in higher frequencies, especially noisy consonants like /f/, /s/ or /h/. Depending on the consonant, the most spectral energy lies above 3 kHz or even 8 kHz. Consonant's spectral energy is often widely distributed and thus occupies a broader bandwidth than a vowel's energy, which is usually concentrated below 3 kHz. [Sha00]

2.2 Language

By natural languages we mean languages evolved naturally and spontaneously in human interaction, expressed in spoken or written form. Other languages, like mathematical, formal and computer languages differ significantly from natural languages by their structure, complexity, expressiveness and development over time.

It is still not completely understood how language is structured in the human brain and how language is acquired by infants. For humans, segmenting speech to meaningful units such as words and syllables, and understanding the meaning comprised of these units is a very easy task and human infants learn this ability rather effortlessly during the first years of their lives. Explaining the mechanisms of natural languages requires extensive knowledge in cognitive processing and brain functions as well as knowledge in speech processing and linguistic theory. [Räs07]

The theory of universal grammar [Coo07], however, suggests that there is a common underlying structure or set of rules, which every natural language grammar is based on. This implies that the universal grammar is embedded in the brain circuitry, and humans conform to this deep structure when acquiring their first language.

2.2.1 Structure of natural languages

The structure of natural languages can be seen as levels of related elements. According to Jurafsky and Martin [Jur08] these levels include, in the order of growing complexity, phonetics, morphology, syntax, semantics and pragmatics.

Phonetics studies the speech production and how the linguistic sounds are realized acoustically. Phonetics itself plays a rather insignificant role in language processing because examining the pure acoustic content does not give information about the meanings. In speech-to-speech machine translation, however, knowledge of phonetics is important, as it forms the basis for successful, accurate speech recognition and synthesis. Morphology studies morphemes, the smallest meaningful language units. These sub-word units joined together with word stems represent the elementary meanings in words, such as singular or plural. Morphology defines the basis for word formation in natural languages.

Syntax describes how the words need to be ordered to form sentences, i.e. what are the structural relationships between words. The same set of words can have different meanings depending on the syntactic rule applied, or not make sense at all if the syntax was not correct. Semantics includes higher-level knowledge of language: it is the knowledge of the meanings of sentences. Natural languages are full of ambiguities, such as homonyms, synonyms, words that can either be a verb or a noun, etc. A system with semantic knowledge, like the human brain, can infer the meaning of ambiguous sentences from its relationship to other words and sentences. Pragmatics, on the other hand, deals with even higher level of meaning than semantics. It contains knowledge of the speakers' (and the hearers', as well) attitudes, goals, intentions, the emotional content of the message and relative concepts. Pragmatics also studies the use of idioms, sarcasm and other possible situations where a sentence doesn't carry its literal meaning in the designated context. The speaker might also refer to other conversations, experiences, or generally known facts. Because of this understanding pronouns like *'it'* or *'that'* correctly require very complex pragmatic analysis capabilities from the language system. [Sha00, Jur08]

Natural languages can be viewed as highly expressive general-purpose knowledge representation systems. The rich structure, context-dependency and redundancy of natural languages make them significantly different from other automatic reasoning systems. Unfortunately, expressiveness and computational tractability in knowledge representation systems are in contradiction. Therefore, the computational characteristics as well as representational and inferential mechanisms of natural languages still remain relatively unknown. [Iwa00]

2.2.2 Challenges from machine translation standpoint

All natural languages apply some set of rules for word formation, sentence formulation, word ordering and word grouping. However, these rules can't be directly mapped between the source and the target language as they can have significant constitutive differences. A demonstrative example of this is the using of articles and verbs in English and Chinese. Chinese language doesn't have tenses for verbs, nor articles such as '*a*' or '*the*'. Chinese uses the context and auxiliary words to express meanings that articles and tenses carry in English. An automatic Chinese-English translator would therefore have to understand the meaning of the utterance in source language, insert the right articles, and choose the correct verb tense in the translation. There are lots of different strategies for word formation in different languages. Some languages apply prepositions or postpositions instead of adding more morphemes to a single word. In some languages some pronouns can be omitted, which causes additional challenges when translating to a language where they can't. [Jur08]

Human translators apply grammar rules and use dictionaries as an information source in their work. Many times though, the target language does not have exact counterparts for all the words in the source language. Grammar rules can include exceptions, and some phrases and idioms might not follow the rule at all. Dictionaries and grammar rules are therefore not sufficient tools for translation. Furthermore, they can not be directly applied in machine translation, because they are designed for human users. Human brain has an astounding ability to supplement information and thus extract meanings even from flawed or inconsistent sources. Also a strong language model, i.e. the knowledge of the syntactic structure of the language and understanding of context, which enables human translators disambiguate words and phrases easily, is very important in translation work. [Nir87]

Currently a big obstacle in machine translation is the lack of deep understanding of the mechanisms of natural languages and the way human translators process them. Deep understanding is required to overcome problems caused by translation divergences described above. Ambiguity and the need for semantic knowledge of the language make natural language processing essentially different from other data processing. Developing a high-quality automatic machine translator with no restrictions or human intervention requires developing a very expressive, formal knowledge representation system. [Jur08, Iwa00]

Some problematic aspects in machine translation are not directly related to the translation algorithm itself, but the communication between humans and machines. Implementing high-quality speech recognition and synthesis in speech-to-speech machine translation applications is a major challenge (see chapter 3). In addition to speech recognition and synthesis, problems such as sentence boundary detection and parsing a spoken language input arise when we want to translate from speech to speech instead of text to text.

Sentences are logical language entities: one sentence represents one meaningful message. Therefore it is obvious that the input needs to be parsed on a sentence-by-sentence basis, instead of word-by-word. Detecting sentence boundaries is not a trivial problem even when processing text input due to the variety of sentence-ending characters and their ambiguous occurring. In speech-to-speech machine translation, the input is a raw speech signal with absolutely no sentence boundary information. Silence periods can be used as one indicator, but more sophisticated speech analysis methods are required when processing continuous speech. Furthermore, parsing the input sentence by sentence causes significant additional delay. The translation system would have to wait for user A to finish a sentence to even start the processing (including encoding, transmission and decoding in addition to the translation process itself) which also causes some delay before the translated speech signal reaches user B. In a telephone conversation, delays of approximately 500ms will be perceived as annoying and reduce the efficiency of the conversation, as they make the dialog broken and unsmooth [Kit91]. This is a significant issue to be solved when developing truly real time, not close-to-real time machine translation applications.

Spontaneous spoken language has often poor syntax and sometimes even poor semantics. Hence the system needs to be able to handle disfluencies, hesitations, incomplete words and sentences etc. when parsing the spoken language input. [Jur08]

3 Speech Recognition and Synthesis

The importance of speech as a communication method has created the need to capture, store and produce speech by machines. In this chapter, basics of automatic speech recognition (ASR) and speech synthesis (or text-to-speech, TTS) are presented. ASR and TTS enable two-way vocal communication between humans and machines, which makes them mandatory operations in all speech-to-speech machine translation systems.

3.1 Speech recognition

ASR is basically a pattern recognition task: each input utterance needs to be identified and mapped to corresponding text. In theory, pattern recognition for ASR could be realized with a large dictionary of waveform entries labeled with the corresponding text representation. Using a certain metric, the system would find the closest match for an input signal in the dictionary and convert it to text according to the labels. This very straight-forward approach is computationally heavy and extremely memory-consuming, and it can only be considered in applications that use a very limited vocabulary. Even in such a situation, the waveform of a single word can vary significantly depending on the speaker and the context. Thus, more efficient pattern recognition algorithms are needed. [Sha00]

The hidden Markov model, or HMM, approach is described here as it is the dominant pattern recognition approach used in large-vocabulary continuous ASR applications. HMM is a machine learning based probabilistic sequence classifier model. To define HMM, we first need to introduce Markov chains, and extend the model to HMM. An alternative approach, maximum entropy model or MEM, is also briefly described.

3.1.1 Markov chains

Markov chains are statistical models, which perform an analysis on a sequence and compute probabilities for the next unit in the sequence to have certain properties or belong to a certain group. In speech recognition, the Markov chain could for example compute the probability for the next phoneme to be a consonant or a vowel. Markov chain is a sequence classifier: it computes the probability distribution over possible labels, and chooses the best label sequence. A logical formulation of a Markov chain is a weighted automaton, defined as follows:

$Q = q_1, q_2, ... q_N$	a set of N states,
$A = [a_{01}\, a_{02} \ldots a_{n1} \ldots a_{nn}]$	a transition probability matrix A, where element a_{ij} represents the probability of moving from state i to state j,
q_0, q_F	a start state and a final state.

A simple Markov chain presented in figure 3.1 shows how to define probabilities of sequences consisting of symbols *A* and *B*. The probability of moving from the state correspondent to symbol *A*, to the state correspondent to symbol *B* is p_1, and to the same state 1- p_1. The probabilities of transitions from the state correspondent to symbol B are p_2 and 1-p_2, respectively. [Jur08]

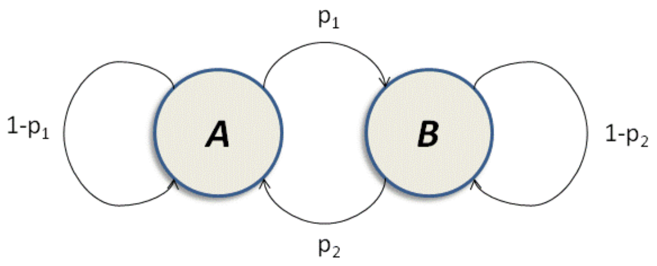

Figure 3.1. A simple Markov chain.

3.1.2 Hidden Markov models

Markov chains can be used to calculate probabilities for events that can be observed in the real world, such as acoustic features of an input signal. In speech recognition however, the most interesting events are not directly observable. For example, in part-of-speech tagging, each word is assigned a tag (noun, verb, preposition, adverb etc.) to decrease ambiguity, and increase the knowledge of possible upcoming words. That is, we need to infer the correct tags from the acoustic source. The tags are indirect, higher-level classifications, which is why they are called hidden. Therefore, the probabilistic model is called hidden Markov model (HMM). HMM is formulated as follows:

$Q = q_1, q_2, ... q_N$	a set of N states, like in Markov chain,
$A = [a_{01}\, a_{02} \ldots a_{n1} \ldots a_{nn}]$	a transition probability matrix *A*,
q_0, q_F	a start state and a final state,
$O = o_1, o_2, \ldots o_T$	a sequence of T observations drawn from vocabulary defined in the model,
$B = b_i(o_t)$	a sequence of observation likelihoods, where each element is the probability of an observation o_t being generated from state *i*.

A simple HMM presented in figure 3.2 shows how the observation likelihoods are connected to the states. For example, symbol *A* in an observed symbol sequence was produced by the underlying process 1 with the probability of 0.6.

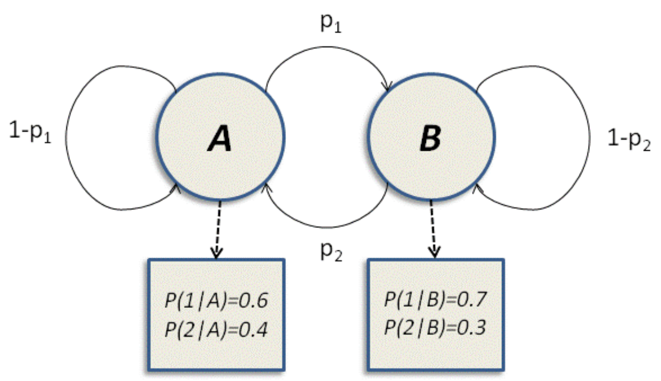

Figure 3.2. A simple HMM.

The observations in a hidden variable model like HMM are inferences, so we can't directly see what was the sequence of variables, that yielded a particular sequence of observations. Finding this sequence of variables is called decoding. The most obvious approach for decoding is to calculate the probabilities for all possible state sequences given the observation sequence, and find the minimum. The large number of possible state sequences makes this approach computationally heavy, and thus not feasible. The most commonly used efficient decoding method for HMM is the Viterbi algorithm. Viterbi algorithm uses a trellis structure and recursive computing to find the most probable state sequence. The observations are processed one by one, ruling out the less probable state paths. See [Jur08] for details.

3.1.3 Hidden Markov models applied to speech recognition

In speech recognition, the word itself can not be directly observed in the real world. We can only observe the acoustical properties of a speech signal related to a specific word. Thus, in HMM-based speech recognition applications, acoustic feature vectors represent the observation sequence, and a sequence of words or sub-word elements represent the hidden states of the model. The task is to solve the most probable sequence of words causing the observation of these particular acoustic features. [Jur08]

Overview of a HMM-based speech recognition system is presented in figure 3.3. As the figure shows, the acoustic features of the input speech signal are extracted and the feature vector is fed into the decoder to obtain the textual representation. The decoder's functionality is based on and trained by two sets of parameters: HMM parameters extracted from a speech database, and language model parameters extracted from a corpus. [Jur08]

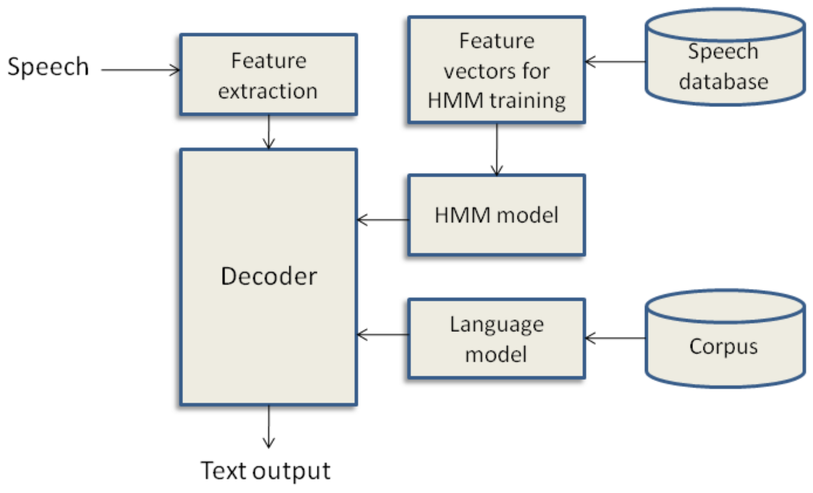

Figure 3.3. Overview of a HMM-based ASR system.

3.1.4 Maximum entropy models

Maximum entropy model (MEM) is another important sequence classifier model. Like HMM, it is a probabilistic sequence classifier and it can be used in speech recognition applications. MEM belongs to the group of log-linear classifiers. [Jur08] It applies the principle of maximum entropy [Jay57], according to which the best describing probability distribution for a phenomenon is the one that maximizes the entropy.

Each speech utterance observation can be seen as a vector of features. MEM processes the input by extracting these features and weighing them by some coefficient factor. MEM is formulated by function $P(c|x)$, which determines the probability for an observation x to belong in class c:

$$P(c \mid x) = \frac{1}{Z} \exp(\sum_i w_i f_i) \qquad (3.1)$$

In (3.1) f_i denotes the features and w_i the corresponding weights. Normalization parameter Z scales the probabilities to sum up to 1. In MEM classification, choosing the features to be analyzed and determining their weights plays a key role. In speech recognition, the features can be for example, morphemes or information of the preceding word. [Jur08]

3.2 Speech synthesis

The task of speech synthesis is to produce acoustic waveforms from text. The challenge is to achieve high quality, natural-like speech output from an arbitrary text input, while keeping memory requirements and computational complexity low. The naturalness of speech consists of various voice characteristics such as speaker individuality, emotional content, or prosodic variation. Producing these features synthetically has proven to be very difficult. Problems can arise from memory restrictions, poor vocoder techniques and inaccurate parameterization. [Rai11, Tok02]

In this chapter, two approaches for speech synthesis are presented. Unit selection approach, based on concatenating pre-recorded speech segments, has been the dominating speech synthesis technique over the past decade [Tok09]. A parametric synthesis method based on HMM models was proposed to improve the accuracy of modeling of speech characteristics, and the flexibility and adaptability of TTS applications. [Rai11, Tok02]

3.2.1 Unit selection

The conventional scheme for TTS implementation is unit selection. In this approach, pre-recorded speech units are selected from a database and concatenated to form words and phrases. The challenge in unit selection is to preserve the natural coarticulation between phonemes while avoiding misalignments of segments during concatenation. The segments to be combined have to meet certain requirements for F_0, energy and duration to minimize disturbances. Furthermore, prosodic variations and intonation contours have to be preserved to achieve natural high-quality speech. [Tor08]

The size and construction of single units, as well as the size of the database varies between different implementations. The size of the database is directly proportional to the phonetic and prosodic context coverage of the system. Larger speech unit databases have better coverage, which allows more control over vocal variations in the speech output. Finding the ideal database size for a TTS system is an optimization problem: with a larger database better speech quality can be achieved, however recording and storing a large database with appropriate variations is difficult and costly. Also, the selection algorithms in large-database applications grow in computational complexity. [Tok09] There’s even more variability in the structure and size of a single unit. Possible realizations are for example diphone, triphone or half-phone units, larger polyphone units, non-uniform or even variable-sized units. Usually, systems using longer units need a larger database to achieve adequate coverage. Smaller units offer more join points and can thus be combined in more versatile ways. However, large number of join points can affect the continuity of speech and cause disturbances. [Tor08, Tok09] Experiments by Torres and Gurlekian [Tor08] show, that speech quality is perceived more natural when a TTS system uses dynamic units with high occurrences of three and four phoneme sequences, compared to the widely used diphone unit realization.

The selection can be performed by minimizing the target cost, which defines how well a particular unit from the database matches the desired unit, and the concatenation costs, which define how well adjacent units combine. The target and concatenation costs are defined according to the equations (3.2) and (3.3), respectively [Tok09]:

$$C^{(t)}(t_i, u_i) = \sum_{j=1}^{p} w_j^{(t)} C_j^{(t)}(t_i, u_i) \tag{3.2}$$

$$C^{(c)}(u_{i-1}, u_i) = \sum_{k=1}^{q} w_k^{(c)} C_k^{(c)}(u_{i-1}, u_i) \tag{3.3}$$

In (3.2) and (3.3), index variables *j* and *k* go through all features in the units under analysis, and each feature is assigned a weight *w*. The features usually considered are F_0, power, duration and formant content of the unit. The weight w determines the relative importance of each feature. [Tok09, Tor08]

Relatively good speech quality can be achieved by unit selection, but such TTS systems are rigid in regards to modeling speech characteristic variations caused by different speakers, emotions, or context of the message. Large databases are required to store the speech data for unit selection, which becomes a substantial memory and computational resource issue, especially in applications with support for multiple languages. The poor adaptability, however, is the main deficiency of unit selection based TTS systems. After the initial setup, adapting the system or training it to acquire new behavior is very difficult or even impossible. [Rai11, Tok09]

3.2.2 HMM-based synthesis

Statistical parametric speech synthesis approach is an alternative TTS scheme growing in popularity in recent years. In contrast to unit selection, statistical parametric speech synthesis aims at generating the average of a set of similar speech segments. A parametric representation is first extracted from a speech database, after which the parameters are modeled, and finally used to generate the synthesized speech. A widely used framework for modeling is HMM. [Tok09] The parameters of HMM can be trained and modified easily. The speech reconstructed from these parameters can thus be adapted to model virtually any vocal characteristics. In regards to adaptability, HMM-based speech synthesis methods clearly outperform the unit selection methods.

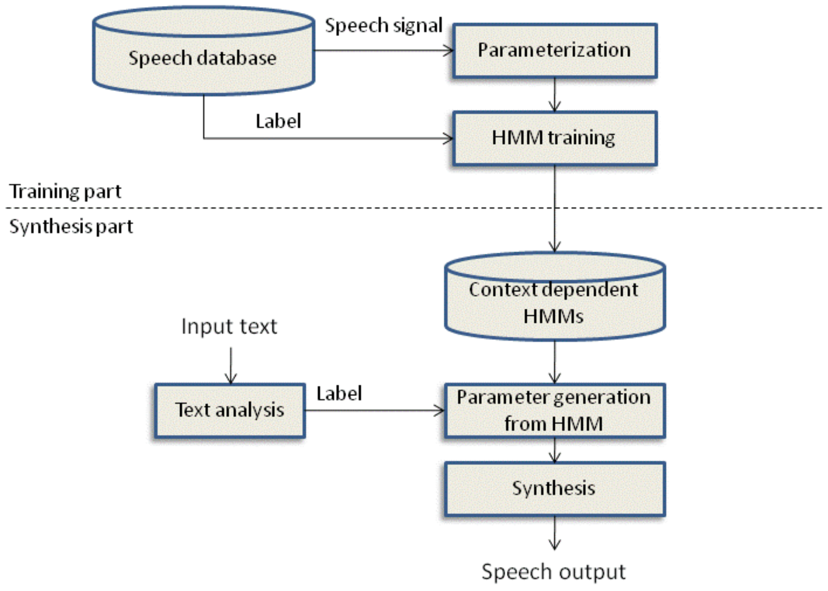

Figure 3.4. Overview of an HMM-based TTS system. [Rai11]

A system such as shown in figure 3.4 consists of both training and synthesis parts. In the training part, each speech utterance from a database is parameterized, and the obtained parameters are trained in a HMM framework. In the synthesis part, the HMM's representing sub-word units are used to generate the needed parameters for synthesis. In most HMM-based TTS systems, the synthesis is based on the source-filter model of speech (see chapter 2.1.1). In a HMM based TTS system proposed by Raitio et. al [Rai11], also the parameterization utilizes an inverse of the source-filter model. The glottal inverse filtering function *G(z)* decomposes the speech signal into the glottal source signal (correspondent to the excitation signal in figure 2.1) and the vocal tract transfer function:

$$G(z) = \frac{S(z)}{V(z)L(z)} \tag{3.4}$$

In (3.4), *S(z)*, *V(z)* and *L(z)* are the z-transforms of the speech signal, vocal tract function and lip radiation function, respectively. This natural glottal source signal and the adaptive parameters from the HMM model are then used in synthesizing the speech waveform. The fine properties of the source signal play a big role in creating the variations of speech that are perceived as naturalness. Thus, using a source signal obtained by glottal inverse filtering yields better speech quality than a simple unit impulse train signal, which is often used as an excitation in source-filter based synthesis. [Rai11]

4 Machine Translation

There are currently a large variety of machine translation (MT) systems, from translation aids with variable degree of human intervention to fully automated translation machines. A fully automated, high-quality translation system with no human intervention, however, still remains a desirable goal.

The methods of machine translation can be roughly categorized in three groups: transfer MT, knowledge-based MT, and statistical MT. In the transfer MT strategy, the source text is parsed and encoded into some abstract meaning representation. This representation will then be “transferred” to the target language by using a lexical-structural transfer dictionary designed for the language pair under analysis. In knowledge-based MT, the use of a transfer dictionary can be omitted by expressing the meaning of the source text using a language-independent meaning representation system also known as interlingua. Statistical MT strategy, however, is an essentially different, machine-learning based approach. [Nir87] In chapters 4.2 and 4.3, the knowledge-based MT and statistical MT strategies will be described in more detail.

4.1 Introduction to machine translation development

The first machine translation technologies were developed in late 1940’s [Nir87, Lop08]. Since then attempts have been made by several universities and research institutes. In the early times knowledge-based translation methods were considered superior to statistical approaches, mainly because of the weaker computational capacity of then-current processors. Statistical machine translation started growing in popularity in the 1990’s, first dominated by IBM and recently also by Google. The progress of machine translation technologies in recent years has been remarkable, but still no application can be used for critical tasks without help of a human translator. However, some current technologies are actually good enough to be used in certain situations, where the quality of the target language style is not the key requirement and translation errors are not crucial. [Gee05, Lop08]

There are a variety of machine translation realizations [Gee05]. The actual product can either be a software package application for a PC, a web service, or a speech-to-speech translation service in a mobile network. Some widely used web-based translation services at the moment are Yahoo! Babel Fish, WordLingo, Bing Translator by Microsoft and Google Translate. At the moment, no widely used, high-quality speech-to-speech translation service is available on the market. Google, having an automatic text translation service covering more than 50 languages, has announced plans to publish a speech-to-speech translation service in 2011 [Eri10a]. Ericsson has developed a working speech-to-speech machine translation service prototype with limited, travel domain scope. This prototype employs an external 3rd party translation engine integrated into the Ericsson network. The service can be run in a normal UTMS network and utilize Ericsson’s existing network components. [Eri10a]

4.2 Knowledge-based machine translation

A knowledge-based approach is the traditional way to view the problem of machine translation. The underlying idea in knowledge-based machine translation (KBMT) is that the grammar of a language can be seen as a theory of structure. The goal is to define a grammar rule that can generate an infinite set of grammatical sentences, based on a finite number of observed sentence utterances, i.e. a corpus. [Cho56]

A common realization is to define a knowledge base, consisting of entity-oriented grammar formalism describing the semantic content of the language, and functional grammar formalism describing the syntax. These together with a lexicon, the set of words and phrases, form a knowledge base. The basic structure of knowledge-based translation system is presented in figure 4.1.

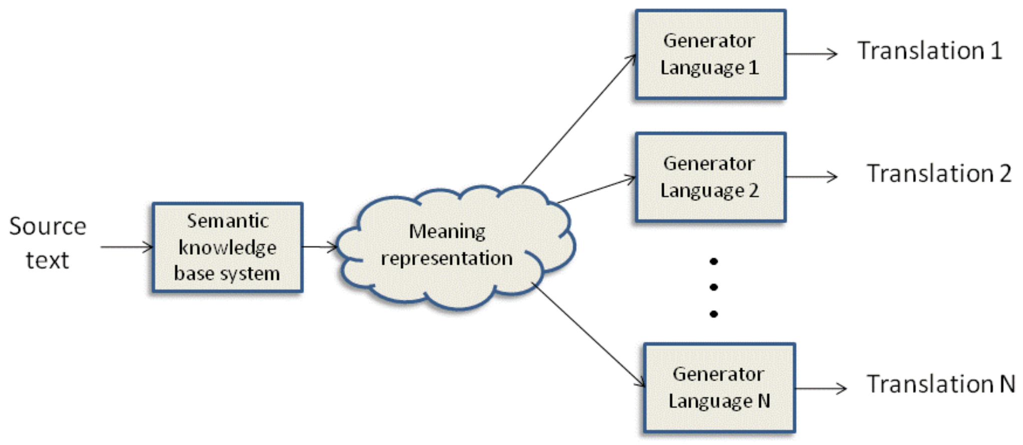

Figure 4.1. A knowledge-based machine translation system. [Nir87]

The process comprises parsing the source text, using the semantic knowledge base to extract the meaning of sentences, encoding it to a semantic meaning representation, or interlingua, and finally generate target language text(s) from the interlingua representation using some formal grammar. The details of these phases are described in the following subchapters. [Nir87]

4.2.1 Interlingua

Interlingua is a language-free, conceptual representation which expresses the meaning of the text to be translated. It is a higher-level notation that can express meanings of natural language without using any syntactic phrase-level or word-level structures of natural languages.

The main attraction of knowledge-based systems utilizing interlingua is the reduced complexity, compared to the transfer MT strategy mentioned earlier in this chapter. *N* being the number of languages known by a translator, a transfer MT system would require N^2 different grammars, one for each language pair. A system using interlingua, however, needs only N parsers and N generators to perform a translation between any two languages. Furthermore, the source language text needs to be parsed only once, and the same interlingua representation can be used to generate translation in any language known by the system. A major drawback of interlingua is that it can only convey the semantic content of a message – in some situations the actual syntactic structure of the text might play a significant role as well. [Nir87]

4.2.2 Formal grammars

A formal grammar is a mathematical system for modeling the structure of natural languages. It describes the syntactic knowledge of a system, i.e. the syntax and the lexicon. In knowledge-based machine translation, formal grammar is used for parsing the source text, and generating the text in target language after translating the meaning. Formal grammars are obviously language-specific; the rules defining sentence structures are unique for each language. However, they are domain-independent. In theory, a knowledge-based translation system defined by a formal grammar could translate any source text, regardless of its genre or domain. [Nir87]

The most commonly used formal grammar is context-free grammar (CFG). It is a logical extension of finite-state languages (FSL), defined in the classical hierarchy of grammars by Chomsky [Cho56]. Finite-state languages can be used for pattern matching by using concatenation, disjunction, union and repetition of symbols of a finite alphabet. CFG has greater expressive power than regular expressions produced by FSL. The formal definition of a CFG is as follows:

N	a set of non-terminal symbols,
Σ	a set of terminal symbols,
R	a set of rules, of the form $A \rightarrow \beta$, where A is a non-terminal and β is a string from the set $(\Sigma \cup N)^*$,
S	a start symbol.

Hence, CFG produces a language consisting of a (finite or infinite) set of strings, i.e. sequences of symbols. In this definition, terminal symbols correspond to words in a language: they are elementary symbols defined by the lexicon. Non-terminal symbols are syntactic variables, or rules that express clusters or generalizations of terminal symbols. In other words, in each rule $A \rightarrow \beta$ in R, A is a symbol that can be rewritten with the string of symbols β. Because β can include both terminal and non-terminal symbols, CFG can define a hierarchically embedded rule set. That is, the lowest level defines the lexicon (i.e. the vocabulary), higher levels express combinations of the symbols in the lexicon and their combinations, producing more and more complex phrase formation rules. [Jur08, Cho56]

Another popular formalism is finite-state transducer, or FST, which is also a generalization of FSL. [Jur08, Lop08] FST is a finite-state automaton, like the automata producing FSL. However, FST defines two sets of symbols instead of one and thus produces two sequences, one from each symbol set. This allows us to look at a FST in several ways. It can be seen as a mapping between two symbol sets, or a machine that defines relations between the two sets. It can also be seen as a machine that reads a sequence from set Σ as an input and produces another sequence from set Δ, and thus acts as a symbol translator. FST is formally defined as follows:

Q	a set of states,
Σ, Δ	sets of symbols,
q_0, *F*	a start state and a set of final states,
δ(q,w)	transition function (or -matrix) defining the set of possible new states, given the current state *q* and an input sequence *w*,
σ(q,w)	output function defining the set of possible output sequences, given the current state *q* and an input sequence *w*.

FST is widely used in problems like ASR and character recognition, and it is a useful formalism for morphological parsing and defining morphological rules, i.e. rules for word formation. FST can read a text input, i.e. a letter sequence, and produce a corresponding sequence of morphemes, i.e. meaningful letter combinations. [Jur08, Lop08]

4.2.3 Sublanguages

In MT systems, sublanguages can be used to inhibit the complexity of a language. Natural languages often have specialized, limited-scope “subworlds” closely related to the domain of discourse. For example, the language used in a particular branch of science, or in a specific kind of media interaction, is a sublanguage. Sublanguages can have specialized vocabulary and even syntax rules.

From MT point of view sublanguages are useful, because the rules for sentence formulation in a sublanguage are more precise. The sublanguage grammars are always smaller than those of general languages. Furthermore, the knowledge of the domain simplifies the problem of disambiguation greatly. Many interpretations, that could be made in general language, are not valid in the sublanguage’s scope. For example, consider the generally ambiguous word ‘shower’ in general language vs. a weather forecast. In the latter case, we can infer the correct interpretation without even seeing the context, but by only knowing the domain of the message.

In a knowledge-based MT system, it is often feasible to analyze the source language input based on both general grammar and a sublanguage grammar. This way the sublanguage grammar can be used as a performance “booster”, to rule out misinterpreted ambiguities or fill in implicit details. [Nir87]

4.3 Statistical machine translation

The growth of the internet together with the increasing number of users in different countries has contributed greatly to the dissemination of multilingual information. The internet has become a huge database of translation references: a countless number of human-translated news texts, government texts etc. are available in electrical form for anyone to access. Together with the fast, cheap computing hardware available today, this has opened up completely new possibilities in machine learning based machine translation development. Statistical machine translation (SMT) has gained interest in recent years and become a considerable alternative for knowledge based systems. Indeed, most modern machine translation applications are based on SMT technology.

The functional principle of SMT is to treat machine translation as a machine learning and probability optimization problem. SMT algorithm is applied on a large database of manually translated texts, often referred to as a bilingual parallel corpus. The algorithm extracts parameters from the corpus, which define a probability distribution. The optimal translation is found by maximizing the probability function and assigning real values to the parameters. Figure 4.2 illustrates the main phases of SMT process, which are translational equivalence modeling, parameterization, parameter estimation and decoding. The phases are described in chapters 4.3.1-4.3.4 in more detail.

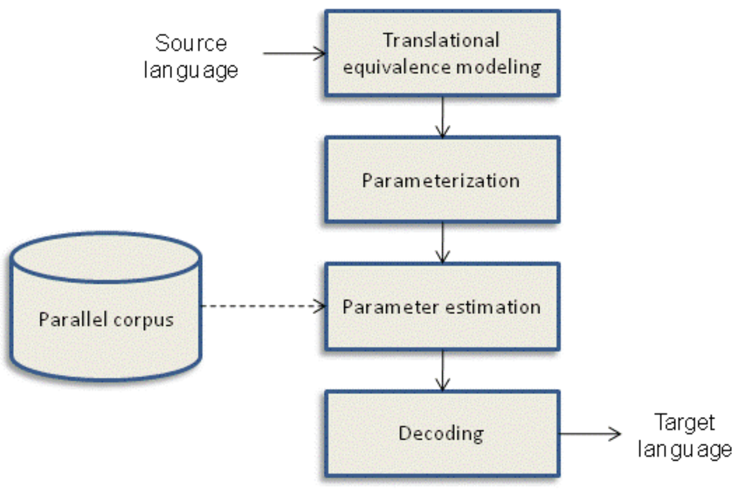

Figure 4.2. The execution phases of a statistical machine translation process.

The SMT approach has a number of advantages compared to the conventional knowledge-based approach. It fits particularly well in multi-language translation systems: no language-specific translation rules, like those in knowledge based systems are needed, and thus the programming work can be omitted as well. This makes it possible to add more languages to system with very little time and human effort. [Lop08, Gee05] As languages are constantly evolving and changing, the MT system has to be agile and able to adapt to the changes. SMT does this automatically when references are replaced or added to the database. The user community can also contribute to MT system performance by crowdsourcing.

4.3.1 Machine learning

Machine learning is the study of computer algorithms, which allow systems to automatically acquire certain functionality through experience, gathered by continuous data processing. A system gets a set of training data as input, and adjusts its weights, or function coefficients, according to the observed features of the training data.

Machine learning can be realized in a supervised or unsupervised manner. In supervised learning, the system is given the target output with the training data. Thus, the system can calculate the difference between the actual output and the desired output, and update its parameters accordingly. In unsupervised learning, the desired output is not provided, but the system tries to find the relevant features of the input data and organize them in appropriate classes. Statistical machine translation can be seen as a supervised learning process, because a parallel corpus contains samples of desired input-output pairs and thus serves as the target output.

In a typical machine learning problem, we introduce the following elements:

- An input data set Y, target set X
- Random variables $\mathbf{y}$, $\mathbf{x}$; ranging over sets Y and X respectively
- Samples $y \in Y, x \in X$

An input data element $y \in Y$ needs to be assigned the best possible output $x \in X$. Sets Y and X can be single or multidimensional. We are interested in the probability function $P(x|y)$, which determines the probability of the output x, when the input y has been observed.

In machine learning problems the task is often to find the best classification for the elements in an infinite input set Y, the set X being a small finite set of labels or classes. In statistical machine translation, however, both sets Y and X are complex, and in case of many languages, infinite. Therefore the machine learning model needs to be extended by a translational equivalence model (figure 4.2), which is basically a set of transition rules. These rules are often derived from theoretical models such as those described in chapter 4.2.2. Because of the complexity of the data sets, the function needs to be parameterized (see chapter 4.3.2) to increase the efficiency and learning capability of the model. [Lop08, Mit97]

4.3.2 Parameterization

Even with a strong translational equivalence model, the ambiguities in the translation process can not be totally eliminated. For each source language (SL) sentence there are still multiple possible target language (TL) translations. The task of parameterization is to create a function that assigns scores to SL-TL sequence pairs. This way we can rank the different SL-TL sequence combinations the model can consider matching. In other words, we want to determine probability $P(e|f)$:

$$P(e \mid f) = \sum_{d:Y(d,e)} P(e,d \mid f) \tag{4.1}$$

In (4.1), *e* denotes the output in TL, *f* the input in SL, and *d* is a rule set based on formal grammars. The rule set *d* yields a set of possible outputs, and the probability *P* is the sum of their probabilities.
Parameterization can be done by using generative or discriminative models. In generative models, the Bayes' rule of conditional probabilities is applied for decomposing *P(e,d|f)*, and thus it can be rewritten as follows:

$$P(e,d \mid f) = \frac{P(f,d \mid e)P(e)}{P(f)} \tag{4.2}$$

However, applying the Bayes' rule forces us to make strong independence assumptions. This can be avoided by using discriminative models, which are based on log-linear modeling. In this approach, *K* feature parameters are defined to create a mapping between the input and the output. Like in MEM, see equation (3.1), the probability can be defined as an exponential function, where the exponent is a linear combination of features *h* and weights λ:

$$P(e,d \mid f) = \frac{1}{Z}\exp(\sum_{k} \lambda_k h_k(e,d,f)) \tag{4.3}$$

Again, feature selection and weight determination is crucial for the accuracy of the model. [Lop08]

4.3.3 Parameter estimation

In parameter estimation, actual values are assigned to parameters in the function *P(e,d|f)* defined in previous phase. Parameter estimation is based on the theoretical assumption, that there is a set of true but unknown parameter values, which together with the language model produces the parallel corpus. The task is to extract the values from the parallel corpus so that they correspond to the assumed true values as closely as possible.

Parameter estimation in generative models comes down to a maximum likelihood estimation problem. The goal is to find a parameter set that maximizes the objective function, i.e. the maximum likelihood estimate. In discriminative models, the main focus of parameter estimation is in determining the feature weights λ. However, if the features were defined using some generative models, the parameters for those models have to be estimated first in the fashion described above. [Lop08]

4.3.4 Decoding

Decoding is the final phase of a SMT process, where the actual translations are produced by using the language model and the parameters estimated in previous phase. The decoder algorithm searches the space of possible outputs and finds the best translation. Due to the large number of possible translations, the main challenge is to find a decoding algorithm capable of searching the space efficiently enough. [Lop08]

A popular framework for decoding in SMT is the stack decoding algorithm presented by Wang and Waibel [Wan97]. The algorithm constructs a hypothesis *H*, including the length of the sentence to be translated and the prefix words of that sentence. The algorithm defines a probability score for the prefix and extends the hypothesis recursively until it completes the sentence. The execution of the algorithm includes the following phases:

1) Initialize the stack with a null hypothesis
2) Pop the hypothesis *H* with the highest score off the stack.
3) If *H* is a complete sentence, output it and terminate the execution of the algorithm.
4) Extend *H* by adding one word from the lexicon, calculate the new score and insert it to the stack. Repeat the same procedure with each word in the lexicon.
5) Return to step 2.

Many SMT decoding algorithms are based on the stack decoding framework and differ from each other for example by the implementation of scoring and stack search. Some algorithms use aborting search, which kills hypotheses under some threshold score, and some apply multiple stacks.

Another approach is synchronous CFG (or SCFG) decoding, based on the CFG language model presented in chapter 4.2.2. The goal is to combine word sequences by using CFG grammar rules, to cover long sentences and finally blocks of text. In SCFG decoding, a sequence of words can be interpreted as a non-terminal symbol. A state in SCFG decoding algorithm consists of a word sequence, the corresponding non-terminal symbol representation and a language model needed to combine different sequences. SCFG finds the best-scoring tree structure that produces the source sentence using the source language grammar; this tree can then be read in reverse order to produce the target sentence in target language grammar. [Lop08]

4.3.5 N-grams

N-grams are word prediction models that can enhance SMT performance especially in noisy environments, or when the source language input is flawed.

According to [Lop08] a word in a source text is “conditionally independent of all but the *n*-1 preceding words”. This means that some word sequences, the length of *n*, have much higher probability of occurring in a text than others. Consider the two following sentences with the same set of words (example originally presented by [Jur08]):

"All of a sudden I notice three guys standing on the sidewalk"
"on guys all I of notice sidewalk three a sudden standing the"

The latter sentence has practically zero probability of appearing in a text, while the probability of the former sentence is clearly non-zero. This way we can compute the probability of the next word in a sentence, and thus predict the word based on the observations of the previous *n*-1 words. The variable *n* defines the length of a sequence under analysis. Thus, *n*-gram is a sequence of *n* words. [Jur08]
N-gram models compute the last word of an *n*-gram from the *n*-1 preceding ones. One approach is relative frequency count based on corpora. In this method we examine a sequence X_1 of *n*-1 words, with another sequence X_2 which contains sequence X_1 followed by one additional word *w*. The equation (4.4) will therefore tell us the probability of word *w* occurring after a certain sequence of words:

$$P(w \mid X_1) = \frac{count(X_2)}{count(X_1)} \tag{4.4}$$

This approach is only theoretically applicable when the vocabulary of a translation system grows and the corpora used for frequency counts get larger. Furthermore, natural languages are not fixed and new sentences are created constantly.
A more sophisticated way of predicting word *w* after a certain sequence is to combine the joint probability of the words in sequence X_2, and the conditional probability of word w given the sequence X_1. The joint probability describes the probability of words occurring together in a sequence, whereas conditional probability describes the probability of word *w* occurring after observing X_1. This can be formulated by using the chain rule of probabilities:

$$\begin{aligned} P(w_1^n) &= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1^2)...P(w_n \mid w_1^{n-1}) \\ &= \prod_k P(w_k \mid w_1^{k-1}) \end{aligned} \tag{4.5}$$

In (4.5), w_1^n is a sequence of *n* words.

N-grams typically apply a small value of *n*, commonly *n*=2 (hence called a bigram) or *n*=3 (a trigram). The small value of *n* allows us to approximate the probability of the word *w* by the last few words instead of its entire history. Thus, we can rewrite the bigram probability as in equation (4.6):

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-1}) \tag{4.6}$$

and further generalize the approximation for the general case of *n*-grams:

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-N+1}^{n-1}) \tag{4.7}$$

This is practical considering MT systems: natural languages are constantly evolving, and one particular sentence to be translated might never have occurred in a corpus. Especially in long and complex sentences, the bigrams or trigrams of a word, however, have occurred in the corpus under analysis with a much higher probability. [Jur08, Lop08]

5 Evaluation

Despite the recent progress in MT technology, the quality of the translations is often far from perfect. The quality is crucial especially when the target users do not know the source language and thus have little, if any, means to recover from possible translation mistakes. The mistakes vary in severity: some of them are slight style errors, some can alter the whole meaning of the message or turn it into complete nonsense. Depending on the severity and type of errors, the discussion may result in amusement, confusion, offense or total mutual incomprehension. Evaluation of the MT systems is therefore indispensable. It serves both researchers and the end users of MT systems. [Tom03, Alb09, Sta10]

Currently there is no efficient, consistent and easy MT verification method. One of the biggest reasons for this is that quality of a translation is a subjective measure. Furthermore, the whole process and its actual goal, correctness of the translation, is difficult to formulate. Even so, considering the importance of verification of MT systems, it is necessary to formulate some kind of metrics for MT quality. [Jur08, Tom03]

The goal is to be able to measure the translational equivalence between the source and the target language. Translational equivalence includes conveying the source language meaning and maintaining the fluency of expression in the target language. Traditionally this evaluation has been carried out by human translators, but there is a strong desire to be able to formulate quantitative metrics capable of measuring translational equivalence. [Lop08] The methods for evaluating the performance of a MT algorithm can be divided into subjective and quantitative methods, which are presented in chapters 5.1 and 5.2, respectively.

The evaluation of the end-to-end performance of a speech-to-speech MT application also involves speech quality analysis. We are interested in verifying the effects that possible variations in the quality of transmitted speech has on MT quality and comprehensibility of the message. The goal is to find a method to measure speech quality so that the result can be compared with the perceived translational equivalence of the transmitted signal. This way, the tests can show the relation between speech quality and MT quality. Some possible approaches for realizing the speech quality analysis for MT systems are presented in chapter 5.3.

5.1 Subjective evaluation

At present, the most reliable way to test a speech-to-speech MT system is subjective evaluation performed by human translators. Because of the linguistic knowledge humans possess, subjective evaluation reflects the actual, true quality of the translation. However, such method is very dependent on the individual performing the evaluation, and the ratings of different judges might vary greatly. It is also difficult to define how to rate long sentences consisting of both correct and incorrect parts, and how to measure the impact of a single sentence's rating on the whole message. Furthermore, subjective evaluation is not feasible from a system test point of view due to high evaluation costs, i.e. time and effort. [Tom03] Despite the costs and subjective results, human-performed or human-aided evaluation is still an important strategy due to the lack of extensive automatic evaluation metrics.

Speech-to-speech machine translation is a multi-stage process where each step is non-trivial. To spot the origin of the problem, MT evaluation needs to be broken down to correspond to different parts of the process. A common way to do this is to examine the ASR result in the source language, the translated sentence in target language (later referred to as forward-translation), or the same sentence translated back to source language (later back-translation).

The current ASR systems' performance is not yet at a reliable level and the systems are very easily confused by different speaker characteristics. Performing a speech synthesis on the word sequence resulting from ASR is a way to prove that the MT system is actually trying to translate the sentence the user intended. However, knowing that the ASR output was correct is only the first step in MT evaluation, and it doesn't tell us anything about the correctness of the actual translation. In the following subchapters, some strategies for subjective rating of the translations are presented.

5.1.1 Subjective scoring

In forward-translation evaluation strategy, a bilingual expert will assign scores to translated sentences using some suitable scale. The well-known Likert scale from 1 to 5 is a commonly used alternative for this purpose. The Likert levels can be interpreted for example as follows by Stallard et. al [Sta10]:

5 – Essentially a perfect translation

4 – An adequate though somewhat disfluent translation which conveys the meaning of the utterance

3 – A partial translation which is missing one or more concepts, or is severely disfluent

2 – A translation which is missing most of the concepts

1 – A translation with no apparent relation to the input.

The forward-translation evaluation strategy is reliable, but time-consuming and requires a highly skilled bi- or multilingual expert to perform the tests. The subjective nature of the task might skew the results, unless the test is performed by many individuals, with the results normalized and averaged afterwards.

In back-translation evaluation strategy, the confirmation utterance is created by running the sentence under analysis through the MT systems two ways: from SL to TL, and back to SL again. In case of an ideal MT system, the back-translation and the SL sentence would be identical. With current MT systems this is of course not always true. However, if the back-translation is correct, the probability of the forward-translation being correct is high. This enables us to utilize back-translation in MT evaluation, and perform the testing even without knowing the target language. A useful approach is also to examine the correlation of Likert scores of the forward- and back-translated utterances. [Sta10]

5.1.2 Information error rate

Information error rate (IER) attempts to find a solution to the problem of rating long sentences consisting of both correct and incorrect parts. In IER approach, the sentences are segmented into groups of one or more words, each representing an entity that conveys a piece of information. These segments are called information items, and each item in a translated sentence is ranked by marking it with a label corresponding to its quality: “OK”, “error”, “syntax”, “meaning”, “others” are examples of possible labels. The IER score can then be calculated as the percentage of poorly translated items, i.e. those not marked with the label “OK”. This way, incorrect information has the same impact on the evaluation score regardless of the length of the sentence where it occured. [Tom03]

5.1.3 Task-oriented evaluation

Another possible approach for subjective MT evaluation is task-oriented: given machine-translated instructions, a person’s performance in a designated task would be measured. [Lop08] Levin et. al [Lev00] proposed an evaluation method based on the idea, that literal translation of the source message words is rarely needed, if the interpretation is correct. The task-oriented evaluation method introduces domain actions (DA), which consist of speech acts and domain-specific actions. Speech acts are general actions, with which words in a message can be associated, e.g. “acknowledge”, “give-information”, “accept”. Domain-specific concepts represent the vocabulary of the discourse. For example, in the travel domain the tags could be “price”, “room”, “flight”, or “availability”. The translated message will then be tagged by a human translator, and the translational coverage can be calculated as the percentage of sentences that can’t be associated with any DA.

5.1.4 Relative weighting

When evaluating machine translated messages, a false acceptance, i.e. falsely rating a translation correct, should be considered worse than a false rejection because of the nature of the consequences: confusion, aggression etc. [Sta10]. The accuracy of evaluation can be increased by assigning weights to the subjective ratings. Relative weighting can be used together with another subjective measure, e.g. Likert scoring or IER, to increase the importance of the crucial parts of the message and diminish the risk of a false acceptance. For example, the presence of certain concepts can be weighted higher in a translated sentence, and style errors can be considered less crucial than serious syntax errors.

5.2 Quantitative methods

The main advantage of quantitative translation evaluation methods is the low cost. After the initial effort, which includes formulating a suitable automatic translational equivalence metric and setting up the test system, the evaluation cost is practically null. The quantitative evaluation approach is, from system test point of view, the only feasible option due to the high volume of tests and the need for frequent, cyclic regression testing. [Lop08, Tom03, Pap02]

To obtain truthful evaluation results, the quantitative metrics need to correlate to human judgement as closely as possible. Implementing this correlation by using an efficient, accurate algortihm is the main challenge in developing a quantitative MT evaluation metric. Some metrics that can be used for automatic evaluation are word error rate (WER), sentence error rate (SER) and BLEU score develop by IBM. These metrics use human translated material as references with which the machine translation output is compared. [Tom03] A metric referred to as METEOR has been developed based on BLEU to improve the evaluation performance [Lav07].

5.2.1 Word and sentence error rate

WER describes how many words in the translation need to be inserted, deleted or replaced to transform it into the corresponding reference sentence. WER is computationally efficient, reproducible and fully automatic. It has been successfully used for evaluating ASR performance, but from MT evaluation point of view it has a number of problems. It can consider only one correct reference translation per one output sentence, and it classifies all word re-orderings and synonymic word choices as errors.

SER is a similar metric to WER, but it calculates the percentage of sentences in a block of text that do not match exactly with the reference. Like WER, SER is not optimal for evaluating MT systems with wide scope and large vocabulary. The results of both WER and SER depend highly on the style and vocabulary of the used reference. This does not necessarily tell us anything about the comprehensibility of the translation.

WER and SER metrics can be extended to use multiple references, which allow the system more variation in the word choices and orderings. In this approach multiple reference sentences are compared with the MT output, the distance between each candidate-reference pair is calculated and the smallest one is chosen. Applying multiple references on WER or SER requires greater initial effort than the basic, single reference approach. However, by applying multiple references, the most significant shortcomings of WER and SER can be eliminated and the test coverage can be improved remarkably. [Tom03, Lop08]

5.2.2 BLEU score

Bilingual evaluation understudy (BLEU) is a more sophisticated method and the most widely used automatic metric for MT evaluation [Pap02]. BLEU compares not only single words (like in WER) or full sentences (like in SER) with the translation. Instead, it uses several references and combinations of different word sequences when estimating the quality of a translation. BLEU also assigns lower rates to translations which are significantly longer or shorter than the reference.

BLEU employs modified *n*-gram precision models (see chapter 4.3.5 for introduction of *n*-grams). The standard *n*-gram precision model counts the number of position-independent *n*-gram matches between the reference and the translation. This is what BLEU does too, but in a modified precision manner. In a modified precision model, the maximum number of times a word occurs in a reference is first counted. Each time the word is found in the translation, the word is considered exhausted and thus can't be matched again. Some MT systems tend to overgenerate common words (such as *'the'*) which produce high precision, but low comprehensibility. The maximum reference count makes sure that BLEU does not give high precision ranking for these kind of sentences. The example presented by [Pap02] illustrates the superiority of the modified precision. Consider the following word sequences:

Reference: *The cat is on the mat.*
Translation: *the the the the the the the.*

In a standard unigram (1-gram) precision model, the translation would be inappropriately ranked a precision of 7/7. Every word in the translation can be found in the reference, but the sequence itself is nonsense. The modified unigram precision in this case would be more truthfully 2/7, because the word *'the'* occurs in the reference two times.
The basic evaluation unit of BLEU metric is one sentence. BLEU modified precision score p_n can be calculated for block of text including several sentences as follows:

$$p_n = \frac{\sum_{c \in \{Candidates\}} \sum_{n-gram \in C} count_{clip}(n-gram)}{\sum_{C' \in \{Candidates\}} \sum_{n-gram' \in C'} count(n-gram')} \tag{5.1}$$

$$count_{clip} = \min(count, \max\_ref\_count) \tag{5.2}$$

To obtain p_n, the number of *n*-gram matches is first computed for every sentence separately. The clipped *n*-gram counts are summed and divided by the number of *n*-grams in the test corpus. BLEU score can combine different *n*-gram precision scores, usually uni-, bi-, tri-, and quadrigrams, by calculating the geometric mean of different scores. High unigram precision score seems to correlate with the adequacy of the translation, whereas longer *n*-gram precision scores indicate fluency.

The length penalty included in BLEU score is calculated over the whole test corpus. The effective reference length is obtained by finding the best match for each sentence and summing their lengths.

BLEU score correlates better to human judgement than the metrics presented in chapter 5.2.1. Two different, but equally good translation candidates with different word choices and word ordering are ranked at approximately the same level. Random word sequences, that don't form any understandable natural language sentence, are not ranked high even though the set of words is the same as in a correct translation. [Tom03, Pap02, Lop08, Jur08]

5.2.3 METEOR

Metric for evaluation of translation with explicit ordering (METEOR) is a metric developed based on BLEU score. It was designed with the goal of improving the performance, especially the correlation with human judgement of MT output quality on sentence level, compared to that of BLEU metric. According to measurements [Lav07] METEOR clearly outperforms the BLEU metric in the sense of measuring the adequacy and fluency in currently supported languages (English, German, French and Spanish).

METEOR metric creates a word alignment between the MT output and the reference translation. This alignment is produced by three different word mapping modules called the exact module, stem module, and the synonymy module. In METEOR one word in the translated sentence can only map to one word in the reference sentence. The exact module maps the words that are exactly the same. The stem module maps the words that are the same after using a special stemmer (for example, in English Porter Stemmer [Por06] is used). The synonymy module maps words that are considered synonyms; synonymity is defined in WordNet resource [Mil10].

The word mapping modules define all the possible word matches between the MT output and the reference, of which the maximal cardinality alignment is chosen. After this, METEOR calculates the score for the sentence pair. The score includes unigram precision *P*, unigram recall *R*, and parameterized harmonic mean F_{mean}:

$$P = \frac{m}{t} \tag{5.3}$$

$$R = \frac{m}{r} \tag{5.4}$$

$$F_{mean} = \frac{PR}{\alpha P + (1-\alpha)R} \tag{5.5}$$

In (5.3) and (5.4), *m* is the number of mapped unigrams found between the sentence pair, *t* is the total number of unigrams in the translation and *r* the total number of unigrams in the reference. In (5.5), the parameter α controls the relative weight of precision and recall.

Because values of *P*, *R* and F_{mean} refer only to single word matches, METEOR calculates a penalty score *Pen* for word order fragmentation, using the fragmentation fraction *frag* as follows:

$$frag = \frac{ch}{m} \qquad (5.6)$$

$$Pen = \frac{\gamma}{frag^{\beta}} \qquad (5.7)$$

In (5.6), *ch* is the number of the fewest possible matching unigram sequences between the sentence pair, (“chunks”), where the unigrams between each “chunk” are adjacent. Variable *m* is the total number of unigram matches. In (5.7), the parameter *β* governs the penalty shape as a function of fragmentation, and the parameter *γ* describes the relative weight assigned to the fragmentation penalty.

The values of the three parameters *α*, *β* and *γ* used by METEOR metric are determined experimentally [Lav07]. The parameters are tuned based on several data sets, with the goal of optimizing them so that the correlation with human judgement of MT quality is maximized.

The implementation of a METEOR-based evaluation system requires both language-specific word matching modules and language-specifically tuned parameters, leading to elevated initial costs. Therefore, to reduce the effort of expanding METEOR to support new languages, some versions of METEOR are restricted to utilize only two word mapping modules. [Lav07]

5.3 Speech quality effects on translation

Transmission in telecom networks can degrade a speech signal in various ways. Disturbances caused by packet drops, filtering, delays, background noise or low bit rate coding can be perceived as buzzing, clicks or breaks in the received speech signal. Poor quality makes the conversation annoying and sometimes even unintelligible. Therefore, verifying the speech quality in telecom networks is important. This can be done in several ways.

A frequently used rating method for speech quality is a subjective, numerical quality indicator called mean opinion square (MOS). MOS is expressed as a single number from 1 to 5, where 5 = excellent, 4 = good, 3 = fair, 2 = poor and 1 = bad. The MOS value is obtained by averaging the ratings given by a number of listeners in standardized [ITU96] listening tests. The drawbacks of MOS evaluation are similar to subjective MT evaluation methods: it is time-consuming and the results depend on the test subjects.

Perceptual evaluation of speech quality (PESQ) is a widely used algorithm, which estimates the perceived quality that would be given to a transmitted signal in subjective listening tests. In other words, PESQ tries to model the test subject by defining perceptual and cognitive models that correspond to human judgement on speech quality. The PESQ score is obtained by calculating the difference between an original signal *X(t)* and a degraded signal *Y(t)*, which is the original signal passed through a telecom network. PESQ was developed to be applicable in end-to-end speech quality tests in telecommunication networks and to eliminate the shortcomings of subjective listening tests. PESQ has proven to correlate strongly with MOS values in most network conditions, excluding some hard distortions such as extreme temporal clipping, side tones, or severe changes in loudness levels. [ITU01]

Bit-exact analysis is an alternative evaluation approach, and its use has increased in automated speech quality testing in recent years. In bit-exact analysis, the desired speech stream and the network output are compared bit by bit. Mismatches such as bit errors in the speech payload, protocol errors, and packet time deviation are reported to the user. Bit-exact analysis method is absolute and accurate, easy to configure for various test setups, and does not require a license like PESQ. It is suitable for monitoring stability tests with high traffic load, multi-party call tests, and most speech codecs and call types. Bit-exact analysis does not take human perception into account, but all mismatches have the same value in the test result. Because of this, bit-exact verification should be done in conjunction with subjective tests. Bit-exact analysis can help to reduce the human effort and increase the accuracy of results in speech quality verification greatly. [Sjö11]

The methods described above have been designed to be used for testing speech quality in communication where translation is not involved. Due to the lack of extensive knowledge of end-to-end verification speech-to-speech MT applications, we don’t know if these methods are suitable for MT verification and whether or not the results reliable. The translation algorithm is a remarkable difference between speech quality testing and speech-to-speech MT quality testing. A MOS value tells us the intelligibility of speech, but does the same value correspond to the comprehensibility of the translation? Can the same methods even be used, when the input signal and actual network output are essentially different signals? The test RTIS test environment designed in this thesis tries to answer these questions.

6 Ericsson Real-Time Interpretation System

This chapter presents Ericsson's implementation of a machine translation system, the Real-time interpretation system (RTIS) service. It shows how the external speech-to-speech machine translation engine is integrated in Ericsson's UMTS network and how the service is set up. The structure and functionality of the RTIS prototype and its components are described.

6.1 Network overview

This chapter describes briefly the UMTS network. UMTS (universal mobile telecommunications system) network is the environment where the Ericsson real-time interpretation system (RTIS) functions. Ericsson's mobile media gateway (M-MGw) is presented in more detail in chapter 6.1.2, as it is one of the key building blocks in running and testing the translation service. The translational functionality resides in an external, 3rd party translation engine, whose algorithms are invoked during a normal call in order to translate the message. The translation engine is presented in chapter 6.1.3.

6.1.1 UMTS network

The basic structure of an UMTS network consists of an access network (AN), a core network (CN), and a user domain. The user is connected to the network by user equipment (UE) and identified by a subscriber identity module (SIM).

The purpose of the access network is to directly contact the UE and to connect the user to the CN domain. The access network for GSM users is called GSM edge radio access network (GERAN), and for UMTS users it is called the universal terrestrial radio access network (UTRAN). The configuration of an access network consists of access network entities, which manage the resources of the AN. In GERAN the entities are base station controllers (BSC) and base transceiver stations (BTS), and in UTRAN base stations (BS) and radio network controllers (RNC). The access network entities are logically correspondent between GERAN and UTRAN.

The core network provides support for network features and telecommunication services, such as management of user location information, control of network features and services, the transfer mechanisms for signaling, and for user generated information. [3GP09]

The UMTS network is logically split into three horizontal layers, each providing different services to achieve more efficient use of resources. The layers are the application layer, the network control layer, and the common connectivity layer. The application layer contains the end-user applications as well as service databases, which operators use to provide services. The application layer is connected to the network control layer via application program interfaces. The network control layer deals with the functionality and signaling needed to provide services between the different types of networks. The network elements in the control layer are called control servers, mobile switching center (MSC) and transit switching center (TSC) in the circuit switched domain. MSC and TSC servers use a specific gateway control protocol to control the common connectivity layer and handle the signaling. No user plane data is handled in the control layer. The common connectivity layer is the backbone of the network, and it handles the user plane data transportation, routing, and switching within CN. It also serves as a gateway between CN and AN, and this task is usually handled by a device called media gateway (MGW). Ericsson's mobile media gateway is presented in the following chapter in more detail. [Wit00]

6.1.2 Ericsson mobile media gateway

Media gateways (MGW) are devices located at the edges of UMTS CN, serving as interfaces to different types of access networks. Ericsson's implementation of MGW is the Ericsson Mobile Media Gateway, M-MGw. M-MGw application is built on a telecommunication technology platform called connectivity packet platform (CPP). CPP forms a distributed multi-processor system including both the hardware and software. CPP provides a real-time control system and support for TDM, ATM and IP protocols. [Eri10b]

Logically M-MGw is divided into several virtual media gateways (VMGW) and resource components, thus allowing connections from several different MSC servers. Figure 6.1 illustrates the functional structure of the M-MGw. VMGW holds the logic for controlling the resource components, which manipulate the media streams. When M-MGw receives an H.248 gateway control message from the network control servers, it delivers it to the correct VMGW and connection handler. The connection handler sets up the new state for connection and allocates needed resources. [Fyr00]

The hardware structure of M-MGw is distributed across different processor boards which specialize in different tasks. The three main processor board types are

- General purpose board (GPB), a general purpose processor containing the main logic for M-MGw and applications running on it,
- Exchange terminal board (ET), a termination board providing interfaces for traffic over different types of transmission lines, and
- Media stream board (MSB), a board specialized in media stream processing tasks using a number of digital signal processors (DSP) controlled by a board processor (BP). [Eri10b]

The actual payload streams flowing through M-MGw are handled and manipulated by the user plane hardware, i.e. ET and MSB boards, and the related software. The user plane includes services and functions for device handling and media stream processing, such as different voice codecs, echo cancelling, DTMF sender/receiver, tone sender/receiver, multi-party call functionality or circuit-switched data service. [Fyr00]

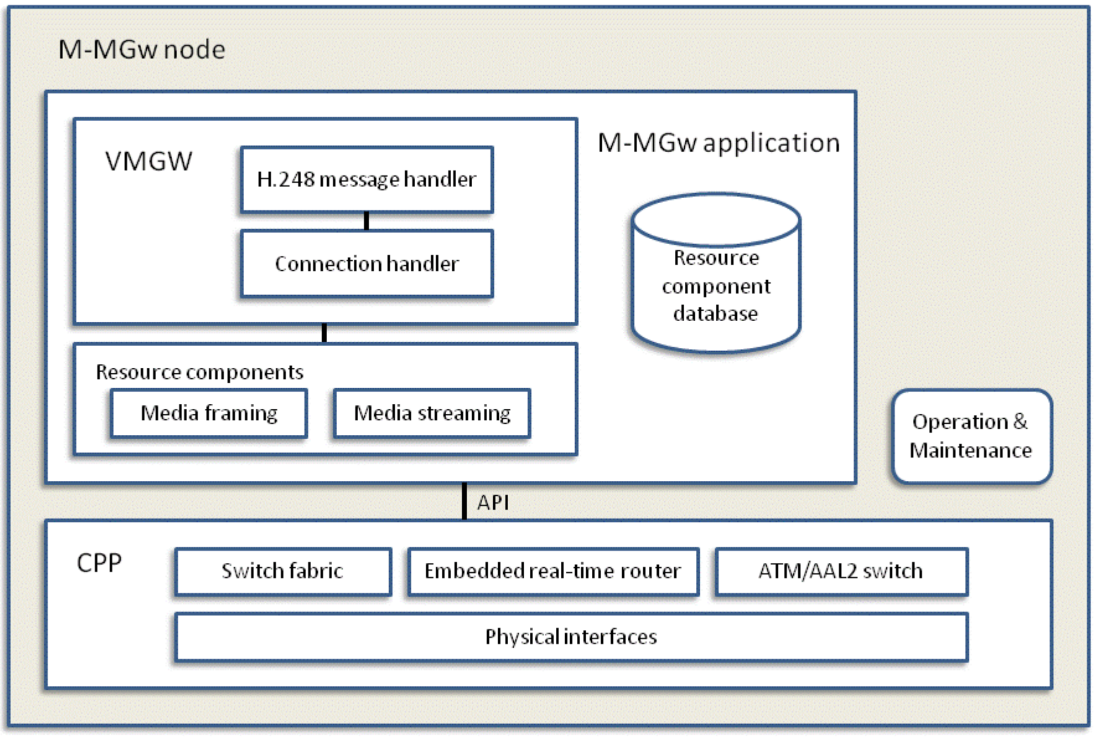

Figure 6.1. The functional structure of a mobile media gateway. [Fyr00]

6.1.3 Translation engine

The 3rd party translation engine used in the current version of RTIS contains the functionality for speech recognition, speech synthesis, and text-based translations. M-MGw sends the recorded speech to the translation engine and invokes its ASR, translation and TTS algorithms. The translation engine sends the translated speech back towards M-MGw, accompanied by the text format representations of the translation and back-translation. Like most modern machine translation applications, the translation algorithms in the 3rd party translation engine are based on SMT (see chapter 4.3) technology. The supported speech codecs at the moment are PCM and AMR.

The translation engine is connected to Ericsson M-MGw over an IP interface. In the future, the translation engine could be integrated in the M-MGw. See chapter 6.2.2 for the current network architecture description. [Eri09]

The information exchange between the translation engine and M-MGw is handled by XML requests and responses. The request and response templates contain fields for attributes such as language ID, user ID, action type (create user request, translate request, status query, translation reply etc.), and content data. Content data holds the actual message to be translated, and the structure of the message is defined in the attributes of

XML requests/responses. A content data document contains text in one language only. A document is represented in XML by the following hierarchy of elements:

- Document, a top level element
- Paragraph, an arbitrary grouping of sentences, containing at least one sentence
- Sentence, a logical leaf element of a document, which may have different representations captured in children, i.e. text elements
- Text, a true leaf of a document structure, present only inside a sentence element

Each sentence element can have none, one, or multiple text children. One text child represents the sentence content from a single source, i.e. either a machine translated sentence or a source language input.

6.2 Real-time interpretation system prototype

6.2.1 Objective

The RTIS prototype was developed to integrate Ericsson M-MGw and the external, 3rd party translation engine. The prototype is designed to work with any mobile phone: 2G, 3G, low end or high end. The goal is to have a real, running, speech-to-speech translation system, which can show how the mobile phone and mobile environment affect the quality of the speech and translations. Some problems may be caused by background noise, network disturbances, transcoding, echo cancellation algorithms, noise reduction algorithms or network delay. Another key question is how does invoking a translation algorithm during a call affect the normal network operations, such as framing or signaling. With the prototype, the service could be appropriately tested to verify the quality of speech and translations. [Eri10c]

At present, the prototype works only in the travel domain, i.e. translation of common travel words and phrases is supported. The current version of the prototype is set up to perform speech-to-speech translations from English to Chinese and Chinese to English, but other languages such as Spanish and Arabic are also supported and can be included in future versions. The development of the RTIS prototype is divided into two stages. The currently working stage 1 supports one simultaneous PCM-coded call, consisting of individual phrases followed by more than 0,5 s of silence. The stage 1 prototype uses one mobile phone in one call. The traffic is routed via the translation engine back to the same phone, which plays out the translated speech signal. In the future, stage 2 will include support for several simultaneous two-way calls, larger language selection, improved accessibility, and more control logic in the user interface. The user will be able to select languages dynamically, and choose if he/she wants to accept the ASR result or cancel and rerecord. The user could also request a back-translation to his/her mobile device in text format. [Eri10c]

6.2.2 Network architecture overview

Figure 6.2 presents the overview of the RTIS service network architecture. The external translation engine is located in the core network and it is connected to M-MGw over an IP interface. To reduce delay, M-MGw and translation engine should reside in the same site.

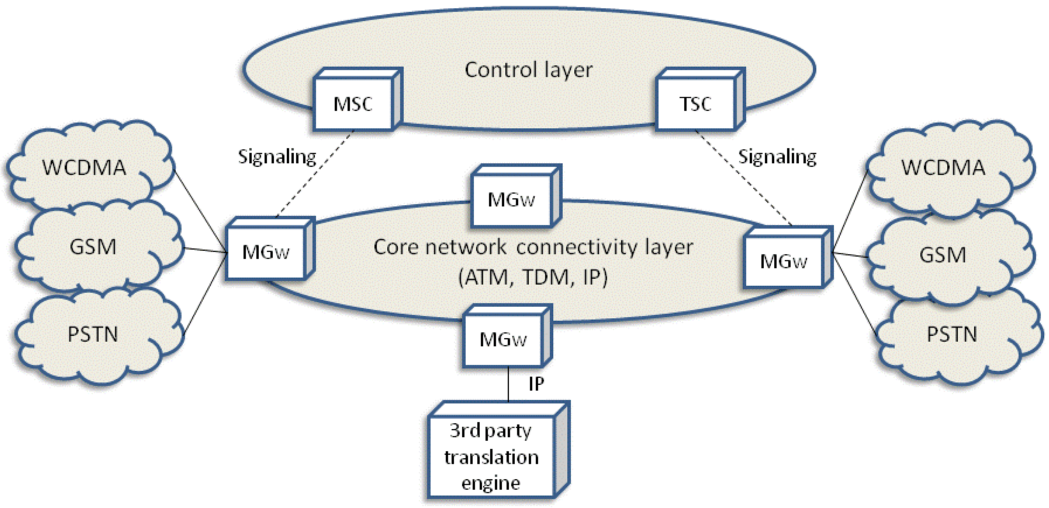

Figure 6.2. Network architecture overview of the RTIS service. [Eri09]

M-MGw supports a number of interfaces, and thus enables connecting and using RTIS service from different types of access networks. Figure 6.3 presents the interfaces in detail. M-MGw connects to the control layer servers (MSC, TSC) via gateway control protocol (H.248). The interface between two M-MGw's is Nb, an open interface standardized by 3GPP. M-MGw supports a TDM interface towards GERAN and PSTN, and ATM, IP and Iu interfaces towards UTRAN. [Eri09]

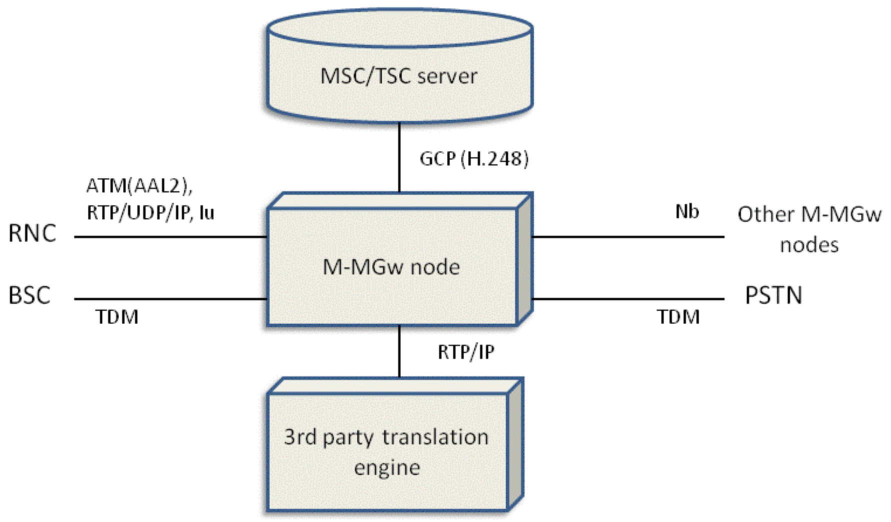

Figure 6.3. M-MGw interfaces in RTIS. [Eri09]

6.2.3 User plane

This chapter describes the RTIS call setup in the user plane. Figure 6.4 shows an example call case, where the call originates from a mobile device operated by an English-speaking user. The Chinese-speaking subscriber B is a landline user. Terminations T1 and T2 in context Ctx1, and termination T3 in context Ctx2 reside in the English domain, whereas termination T4 in context Ctx2 resides in the Chinese domain.

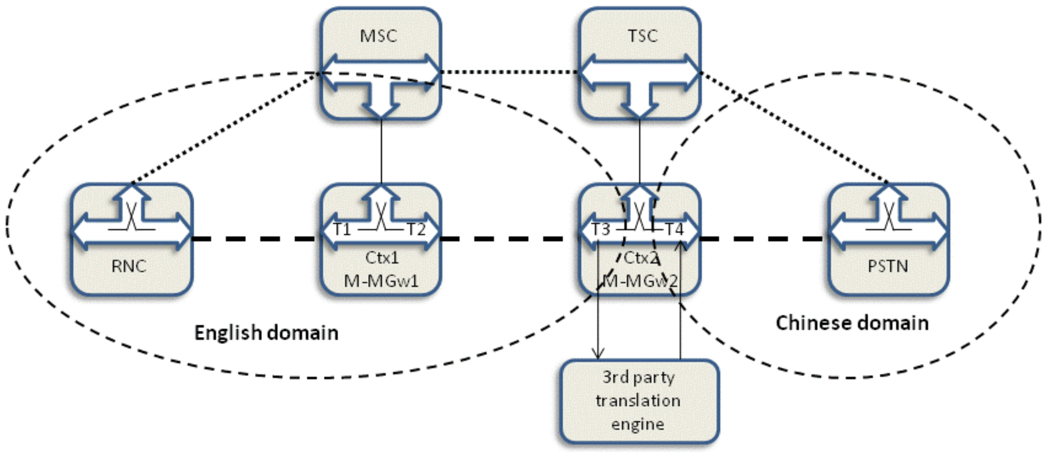

Figure 6.4. RTIS call setup in the user plane. [Eri09]

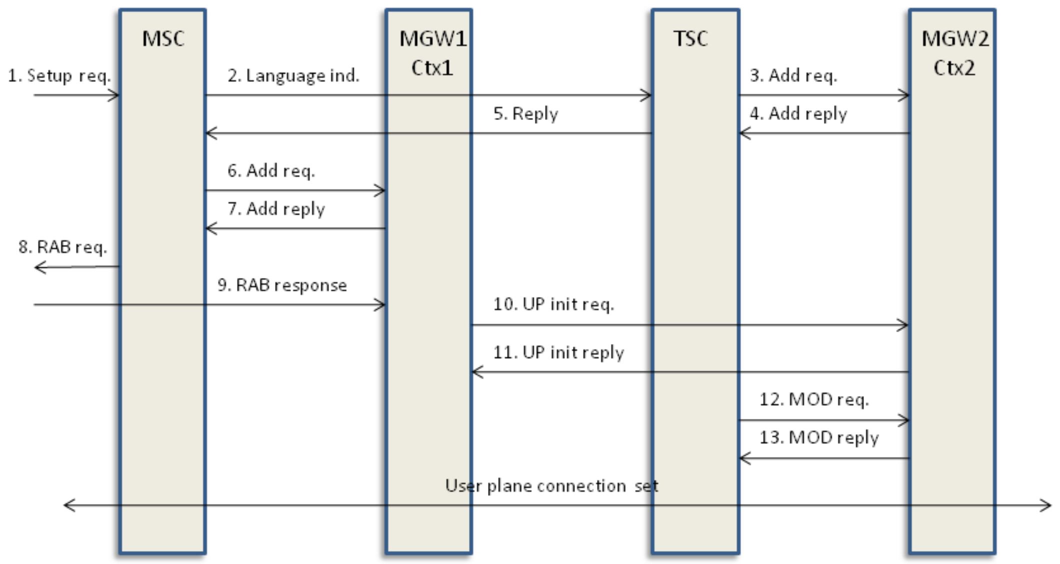

Figure 6.5. Signal flow in a RTIS call setup. [Eri09]

Figure 6.5 presents the setup procedure. The call setup is initiated by a request sent by the mobile subscriber forwarded to MSC (1). In RTIS calls, the control layer needs information of the subscriber's language. A language indicator parameter can be included in the request message, or MSC can choose a preferred language based on the subscriber data. MSC analyses the request message and forwards the language indicator to TSC (2), which handles connections towards PSTN. TSC selects M-MGw 2 to be used and forwards subscriber A's language indicator to that M-MGw (3). M-MGw 2 then creates Ctx2, and returns the M-MGw ID with bearer address and binding reference to TSC (4) and MSC (5). MSC selects M-MGw 1 to be used, which creates Ctx1 (6). M-MGw 1 returns the corresponding bearer address and binding reference information to MSC (7). RNC uses the bearer address to request bearer connection between RNC and M-MGw 1. RNC initializes the Iu UP framing protocol, and M-MGw 1 acknowledges the initialization (8, 9). Now radio access bearer assignment is completed.

M-MGw 1 initializes the Nb UP framing protocol (10), and M-MGw 2 acknowledges the initialization (11). M-MGw 2 informs TSC that the bearer establishment between M-MGw 1 and M-MGw 2 is now complete. However, the language indicator for this termination is not yet known. When B-party alerts, TSC sends a modification request to Ctx2 in order to set the language indicator in termination T4 (12). When B-party answers, a modification reply is sent to TSC and language indicator in T4 is set to Chinese (13). M-MGw 2 then detects, that the language indicators terminations T3 and T4 don't match, and the translation algorithms in the translation server related to M-MGw 2 are invoked. Thus, instead of routing the traffic directly from T3 to T4 within Ctx2, M-MGw 2 routes the traffic via the translation server. Now the user plane connection is up and the transfer of the actual payload can start. [Eri09]

7 Real-Time Interpretation System Verification

As the RTIS prototype is up and running, there is a need to measure its performance and verify future improvements. In this thesis, a test environment for RTIS verification was designed and implemented. Because of the problems related to the subjective evaluation discussed in chapter 5.1, with high evaluation costs being the most important factor, we wanted to study the possibility of automating RTIS verification. The key question is, can an automated test system measure the performance of RTIS, and can it obtain results that correspond to the actual quality perceived by a human user? The goal is to be able to run automated tests as a complementary method in addition to manual testing. This is needed to speed up regression testing and enable system tests with high traffic load, as well as long time period stress tests. In this chapter, the automation strategy and implementation of the RTIS test environment is described.

7.1 Automation strategy

The main goal and purpose of RTIS verification is to show how the network elements and functions affect the speech recognition, translation, and quality of speech synthesis. In other words, we want to measure the end-to-end performance of the integrated network service, not just the standalone translation engine and its algorithms. However, correct translation is still the core and most important element in the RTIS service. This defines the basic requirements for the test environment to be designed:

- The speech data stream should be compared with the desired output on bit stream level,
- It should be possible to include all applicable codecs (e.g. PCM, AMR, AMR-WB, G.729) and services (e.g. echo cancellation, noise reduction) in the test suite in the future,
- A faulty translation should always make the test fail, no matter how accurately the network transmits the signal.

Furthermore, the test environment should be easily combinable with the existing speech quality test setup and preferably use the same test tools. This will ensure the efficient use of resources (e.g. hardware, time, competence) and keep the evaluation costs low.

As a result of the above mentioned, the test environment used for speech quality testing in Ericsson [Eri10d] was used as a basis of RTIS test environment design. Speech quality in telecom networks has conventionally been tested by means of manual listening tests, and automatic bit-exact analysis. In recent years the share of bit-exact verification has increased: currently system verification is conducted in a fully automated manner but some listening tests are still used in development work. In speech quality tests, the bit-exact verification method has proven to be easy to configure, accurate, and feasible in system tests and high load stress tests. On the other hand, the accuracy makes the method prone to errors which can, in noisy environments, skew the results more than e.g. a listening test or PESQ-analysis based test. The goal of designing the RTIS test environment was to investigate, if the bit-exact analysis tools and test methods can be adapted to support RTIS verification, and thus enable the use of automatic tests as a complementary RTIS verification method. The most remarkable difference between speech quality verification and RTIS verification is the translation algorithm: a complex process which changes the content of the transmitted signal in a way that can't be predicted with 100% certainty. Because of this, the speech quality test environment needed to be expanded to include a translational equivalence model, which specifies how the reference translations are defined and how they are compared with the RTIS output.

The strategy for evaluating the translational equivalence was chosen to be multi-SER, an adaptation of SER (see chapter 5.2.1). Multi-SER assigns each input utterance several different fixed reference translation patterns. Each reference is compared with the actual translation received from RTIS, produced by a particular input utterance. If one of the references is a bit-exact match, the test case will pass. If none of the references match, the test case will fail, and the test setup will provide a closeness measure showing the closest match and the severity of disturbances related to that input-reference pair.

In a translation system with limited scope and limited vocabulary, like RTIS, a single input utterance can generate only a limited number of different translations. For example, depending on the current state of the translation engine, a simple input utterance *"hello hello"* can either translate to "你好" (*"Nǐ hǎo"*) or "喂喂" (*"Wèi wèi"*), which are both correct. However, based on the experiments on the RTIS prototype, the input *"hello hello"* never produces "喂你好" (*"Wèi nǐ hǎo"*), which is a combination of the two former translations, and also correct Chinese and an acceptable translation. This suggests that with the current scope and translation capabilities of RTIS, multi-SER with a finite number of fixed reference patterns is a suitable verification approach. Based on the way RTIS produces translation utterances, the assumption can be made that a multi-SER – based verification system can provide an approximately equal coverage than a more sophisticated, BLEU- or METEOR-based verification system, but with less initial effort.

7.2 Implementation

Ericsson's speech quality test environment [Eri10d] was used as a basis for RTIS test environment design. However, a number of modifications to the test environment were required to support RTIS verification. A demonstrative test suite for RTIS was also written during the implementation. In the following chapters, the implementation of the RTIS test environment and the test execution procedures are presented.

7.2.1 Overview

The overview of the RTIS test environment is presented in figure 7.1. The test user controls the system from a Linux workstation using the test notation language TTCN3. TTCN3 opens a FTP- or Telnet connection to M-MGw, and sets up a call by reserving and connecting the needed resources. From M-MGw the call is routed towards the translation server, and the translated speech is sent back to M-MGw for analysis. One call is run and analyzed at a time. Once the call is set up, a predefined input speech pattern is sent to the network and routed via the translation engine. The received speech stream will then be compared with the predefined, expected speech pattern and errors are reported back to the user. A more detailed description of the test tools and resources is presented in chapter 7.2.2.

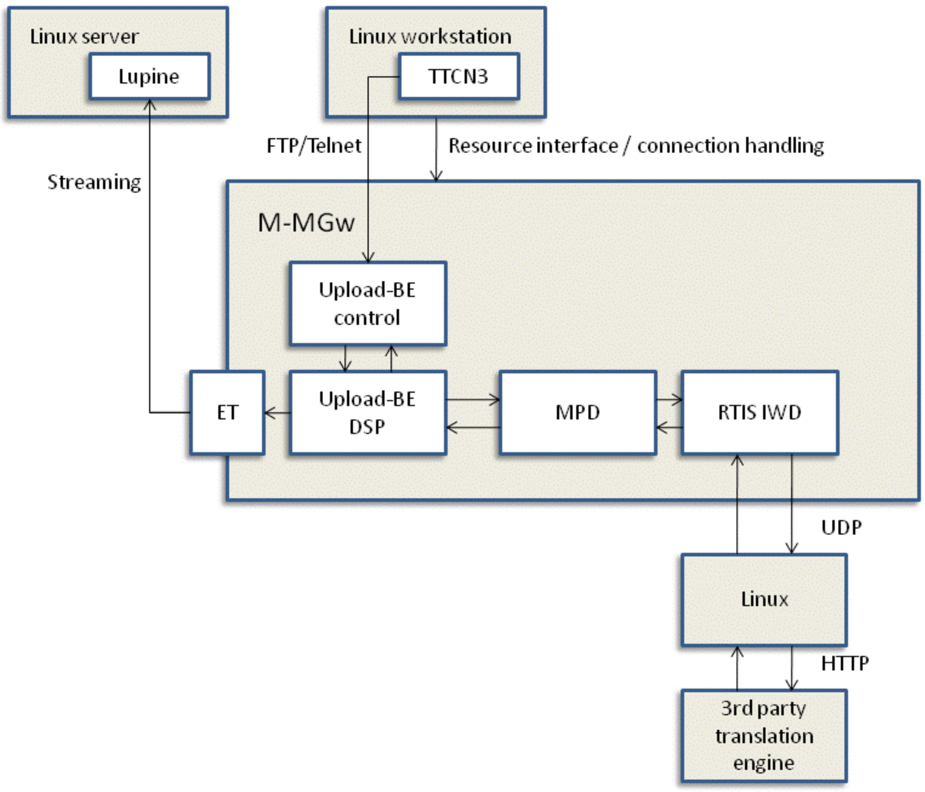

Figure 7.1. The structure of the RTIS test environment.

The changes implemented to support RTIS verification can be summarized as follows:

- Reconfiguration of the test setup. RTIS IWD, translation engine and the interconnecting Linux machine were defined, and new functions for reserving and connecting to these resources were written.
- Defining the translational equivalence model and the principle for rating RTIS outputs.
- Modification of the Upload-BE test tool. A metric indicating the closeness between the received and expected speech streams was implemented.
- Constructing a RTIS test suite. This included formulating a test logic, writing the test cases in TTCN3, creating suitable test data (input- and reference patterns), and running the tests.

7.2.2 Tools

Upload-BE is a user-plane traffic generator with bit-exact analysis feature. It uses predefined speech files stored in DSP memory as stimuli and reference patterns. The stimuli and references can be redefined for each call. Upload-BE supports a number of interfaces, speech file formats and simultaneous calls, and it can be used in testing of calls involving several different network services. The operating principle of upload-BE is illustrated in figure 7.2.

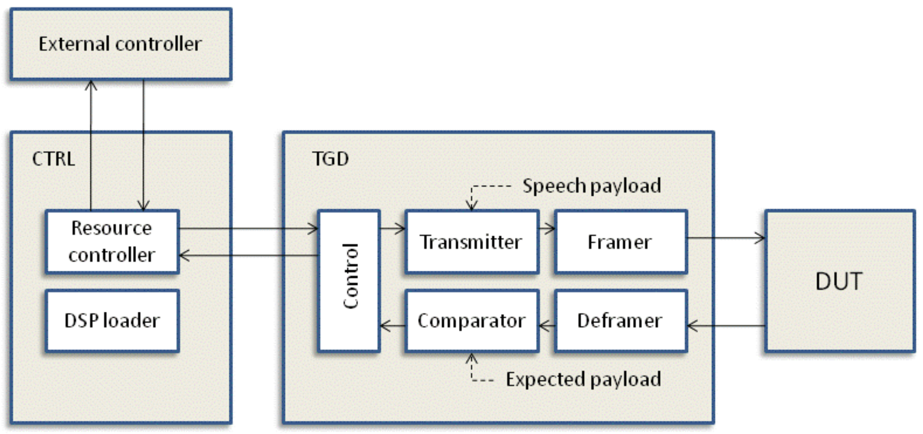

Figure 7.2. Upload-BE operating principle. [Eri06]

Upload-BE consists of a control part (CTRL) and a traffic generator device (TGD) part (see figure 7.1). The control part executes on a GPB board and the TGD on a DSP within a MSB board. CTRL acts as an interface towards an external test controller (in this case, TTCN3), and it forwards the requests to TGD. CTRL is also responsible for loading and configuration of the DSPs, resource selection and statistics. TGD is responsible for framing and sending the speech data to the device under test (DUT), receiving and deframing data, performing the bit-exact analysis and sending notify messages to the control interface. In bit-exact analysis, the speech payload received from DUT is compared bit by bit with the expected payload and mismatches are reported. [Eri06] For the RTIS test environment, upload-BE was extended with continuous analysis and error counter features. Instead of only giving information of a pass or a fail, upload-BE performs analysis on the whole speech stream continuously and counts the total number of defected bytes. This number is then sent to the control interface with the actual analysis result (pass/fail/analysis not performed). Upload-BE also returns information on possible protocol errors, and a timing analysis consisting of maximum, minimum, and average cell deviation time.

Upon the activation of an Upload-BE device, CTRL checks which codec and framing is requested. The framing, if applied, will be started according to the request, and the TGD will start sending a homing pattern. Homing pattern is a pre-defined, fixed pattern, whose purpose is to set the transcoders in the DUT into a known state, and synchronize all the devices in the call chain. In user plane verification, the main challenge is to be able to generate data streams stable enough for bit-exact analysis. Using a homing pattern to initialize the call chain solves this problem and forms the basis of Upload-BE functionality. When all devices in the call have detected homing and synchronization is received, Upload-BE will start sending the actual payload and the bit-exact analysis will start. [Eri06]

One Upload-BE device represents one party in a call. In RTIS test cases, one Upload-BE device is reserved per call, corresponding to the stage 1 of the prototype. In the future, stage 2 can be realized by reserving a device chain such as shown in figure 7.3. In this setup both Upload-BE devices send and receive traffic, and perform the analysis.

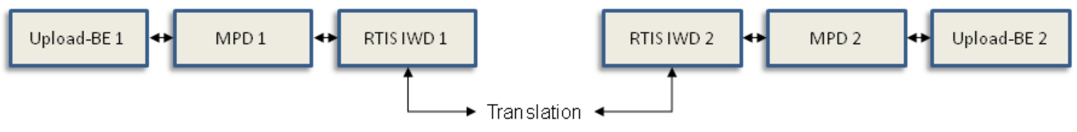

Figure 7.3. Device chain in a two-phone RTIS call.

Media processing device (MPD, see figure 7.1) executes on a DSP and provides speech stream processing functions and services. The parameter sets and resources for running a specific service during the call must be defined and reserved in MPD.

RTIS interworking device (RTIS IWD, see figure 7.1) acts as an interface combining the translation system with M-MGw and the rest of the network. RTIS IWD detects speech activity and starts or stops recording accordingly. It converts the recorded speech traffic to suitable format, frames it into UDP packages, and sends it towards the translation engine. Currently, RTIS IWD supports only raw PCM data with no framing.

The Linux machine between M-MGw and the translation engine receives data in a specific port defined by the test user, connects the M-MGw and the translation engine, and calls the translation functions. The connection process records the received (source language) and sent (target language) speech streams in a file, which can be used to check ASR and TTS results manually. The Linux machine also frames the traffic into HTTP and acts as a firewall.

An IP termination on an ET board (see figure 7.1) routes the traffic towards a Linux server, where Lupine executes. Lupine is a tool for recording traffic for manual checks, i.e. it allows the tester to listen to the actual output speech signal in case of an inconclusive test result. Upload-BE DSP does the streaming if a request is defined in TTCN3.

Test and test control notation version 3, or TTCN3 for short, is a programming language internationally standardized by ETSI, specialized in testing and certification. Due to its wide applicability, the variety of application domains and types of testing, it is typically applied in testing of complex industrial systems, like a 3G mobile network. Compared to conventional programming or scripting languages, the upsides of TTCN3 are the ability to specify dynamic concurrent testing configurations, support for timers, concept of verdicts and a verdict resolution mechanism, and the ability to specify data and signature templates with powerful matching mechanisms. [ETS10] These features suggest that TTCN3 could also be used in RTIS verification.

The TTCN3 core language defines a module, which consists of definition and control parts. The definition part of a module comprises test components, communication ports, data types, constants, test data templates, functions, signatures for procedure calls at ports, test cases, etc. The control part of a module calls the test cases and controls their execution. Local variables and program statements can also be used in control part to specify the execution of test cases. The test cases in TTCN3 define the dynamic test behavior by using program statements such as alternative reception of communication, timer events, default behavior, and test verdict assignment. Template data structures can be used for parameterization and specifying test data to be sent or received over the test ports. [ETS10]

The development suite for TTCN3 currently in use at Ericsson is TITAN TTCN3 Test Executor, TITAN for short. TITAN compiles the test cases written in TTCN3 and generates the code skeleton. TITAN graphic user interface features include smart text editing, compiling, execution of the compiled test suite, test configuration, test monitoring, and log browsing.

7.2.3 Test execution

The main logic of the test execution is presented in figure 7.4. Each test case is assigned one stimulus file and *N* reference translation files. Stimulus is the input for the translation engine. The translated speech is sent back to M-MGw, where Upload-BE performs the bit-exact comparison between the received speech stream and each reference one by one. If the reference under analysis doesn’t match the received speech stream, Upload-BE stores the analysis result and the error counter value in the test case logs. After analyzing all *N* references, Upload-BE returns the number of successful analyses, which can be either 1 or 0, and the verdict is set for test case (pass/fail).

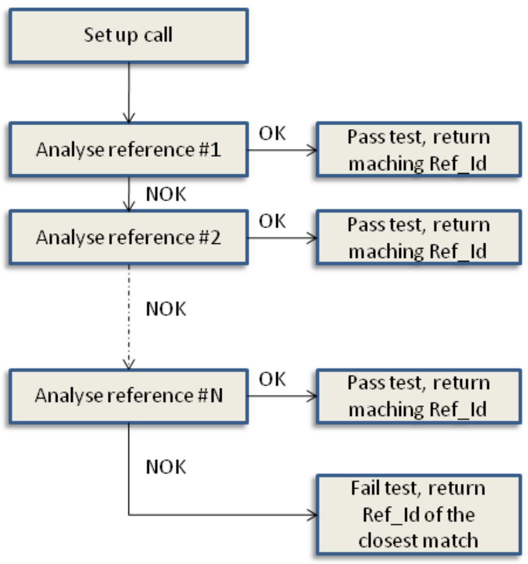
Set up call
Analyse reference #1
OK
Pass test, return maching Ref_Id
NOK
Analyse reference #2
OK
Pass test, return maching Ref_Id
NOK
Analyse reference #N
OK
Pass test, return maching Ref_Id
NOK
Fail test, return Ref_Id of the closest match

Figure 7.4. Logical flowchart of RTIS test execution.

All test cases follow the same basic structure, including 10 main steps:

1) Hunt needed processes, register to GPB and attach to DSP(s)
2) Upload the stimuli and references to the DSP(s)
3) Reserve connection end points
4) Reserve the MPD, RTIS IWD and Upload-BE devices with appropriate service parameters
5) Add terminations to Lupine and start recording traffic
6) Connect the call chain, initialize user plane
7) Send traffic and perform analyses
8) Disconnect call and release devices
9) Check the results
10) Update the test database.

In step 6, the internal connection between MPD and RTIS IWD is established first. The IP connection request is then sent to the Linux machine connecting M-MGw and the translation engine. A connection process running on the Linux machine will forward the data to the translation engine and invoke the translation algorithms. Finally, the internal connection between Upload-BE and MPD is established and Upload-BE will start sending traffic. After completing the analysis of one reference in step 7, a modification request is sent to Upload-BE from TTCN3. The modification signal sets the new reference ID and resets the analysis state. After analyzing the last reference defined in the test case, the execution will move on to step 8.

7.2.4 RTIS test suite

In this thesis, a new group of test cases was created for RTIS verification. The test suite is experimental, and mainly designed for test environment development purposes and demonstration of the concept. However, the test suite can be easily extended to cover the testing of all necessary features in the future. The main functionality of the test cases is defined in TTCN3, including the stimuli and reference files, the number of references, service parameters and interface types. The stimuli and references are loaded to the M-MGw in the beginning of each test case. This way, adding new test cases and modifying the existing ones only requires redefinition of parameters and recompiling of the test suite.

The RTIS test suite consists of 2 test cases. The stimulus utterances were chosen so that they support the demonstrative nature of the test suite. The stimuli are simple English phrases, which are known to be inside the translational scope of RTIS, but on the other hand, are ambiguous in Chinese and have several possible translations. The reference translation sets were compiled based on two criteria: 1) what is correct Chinese and conveys the original meaning of the message, and 2) what are the common phrase structures and word choices in translations generated by RTIS based on experiments. When blocks of sentences are fed directly into the translation engine in text format, depending on the preceding content of the message, the translation algorithm changes its internal state. Thus it can produce translations with variable word choices and syntax structures. However, the experiments show that some words and structures occur more often than others. This information was utilized in choosing representatives from the large number of possible Chinese translations: two references were chosen for test case 1 and 6 for test case 2. The references were synthesized by using the same TTS algorithm as used in RTIS translation engine. The chosen stimuli and reference sets are listed below.

Test case 1:

Stimulus:	“Hello, hello!”
Reference 1:	“你好!” (“Nǐ hǎo!”)
Reference 2:	“喂喂!” (“Wèi wèi!”)

Test case 2:

Stimulus:	“Hello, how are you today?”
Reference 1:	“你好你今天好吗?” (“Nǐ hǎo nǐ jīntiān hǎo ma?”)
Reference 2:	“你今天好吗?” (“Nǐ jīntiān hǎo ma?”)
Reference 3:	”你好你今天怎么样?” (“Nǐ hǎo nǐ jīntiān zěnmeyàng?”)
Reference 4:	“你今天怎么样?” (“Nǐ jīntiān zěnmeyàng?”)
Reference 5:	“你身体好吗?” (“Nǐ shēntǐ hǎo ma?”)
Reference 6:	“你身体怎么样?” (“Nǐ shēntǐ zěnmeyàng?”)

Some references in test case 2 don’t include counterparts for all the words in the stimulus message. For example, references 1 and 2 are essentially the same, except the two beginning characters “你好” meaning “hello”. Reference 2 can still be considered correct, as it conveys the general meaning of the message, i.e. greeting a person. References 5 and 6 don’t contain the word “today”, but are still considered acceptable translations. The core meaning of the stimulus phrase in test case 2 is “how are you”. The word “today” has little importance because, with a very high probability, the point of the question is not to ask how the person feels *today* as opposed to some other day. This holds true especially considering the limited scope of RTIS.

The test cases in the RTIS test suite involve only PCM coding because of the codec restrictions in the current version of RTIS IWD. PCM coding provides high speech quality in normal telephone communication with no translation, and thus is helpful for discovering the possible disturbances arising specifically from the translation service. However, as we are interested in a UMTS network service performance, it is important that the test system can also be used to measure the effects the translation service has on different framings and highly compressed, variable-rate speech, such as AMR or AMR-WB. Therefore, we need to investigate if the same test environment can be used in the future, when support for different speech coding schemes is included in RTIS components.

8 Results

8.1 Results of example test cases

This chapter presents the bit-exact analysis results yielded by the example test cases presented in chapter 7.2.4. Table 8.1 shows the end results with a rating for each reference utterance. As the table shows, no reference defined in the RTIS test suite was rated as correct, and both test cases 1 and 2 failed.

Table 8.1. Bit-exact analysis results and translation ratings of RTIS test cases.

Test case	**Reference**	**Translation rating**	**Final Verdict**
1	1: 你好!	Analysis failed: null	
	2: 喂喂!	Analysis failed: null	FAIL
2	1: 你好你今天好吗?	Analysis failed: no sync received	
	2: 你今天好吗?	Analysis failed: 2649 B (18%) defected	
	3: 你好你今天怎么样?	Analysis failed: no sync received	
	4: 你今天怎么样?	Analysis failed: no sync received	
	5: 你身体好吗?	Analysis failed: 6099 B (42%) defected	
	6: 你身体怎么样?	Analysis failed: 7204 B (47%) defected	FAIL
3	1: 你今天吗?	Analysis OK: 0 B (0%) defected	PASS

The problems observed in the trial runs were caused by the speech activity detection algorithm in RTIS IWD. The stimulus utterance in test case 1 is very short and the activity detection algorithm does not react quickly enough. Thus, no speech is recorded and nothing is sent to the translation module. In test case 2, the duration of the stimulus utterance is longer and RTIS IWD starts recording data. Nevertheless, the speech activity detection reacts too slowly and drops the beginning of the stimulus phrase: only words "are you today" are recorded and sent to the translation engine. This results in an incorrect translation "你今天吗"; it corresponds literally to what was sent, but is neither syntactically nor semantically correct Chinese. The sequence "你今天吗" was included in the reference set for experimentation (see test case 3 in table 8.1), after which the test case passed with 0% defected bytes.

Many references are rated as "no sync received", which means Upload-BE can't find synchronization between the reference and the incoming bit stream, and analysis is not performed. As a result, three references were completely rejected, even though all of them were partially correct.

The possibility of using the bit-exact analysis method to run tests with different speech codecs was investigated by feeding AMR coded speech directly to the translation engine, and examining the output. The trial was done by first feeding in speech without homing, and then by adding a homing pattern in the beginning of the stimulus file. When homing is not used, the RTIS output is not bit-exact, i.e. the same word sequence can yield several different bit streams, depending on the transcoders' state at the time. The purpose of the homing pattern is to initialize the transcoders so that they are in a known state and thus produce an unambiguous bit stream. A homing pattern was included in the RTIS input to verify if this holds true also in RTIS environment. It was observed, that despite initializing the transcoders with a homing pattern, RTIS output is not unambiguous on a bit level. Thus, the bit-exact analysis can't be performed and the tests can't be run in the same way as in case of a PCM call.

Unlike the bit-exact analysis method, PESQ does not have the requirement of bit-exactness between the input and the output. Hence, it is theoretically possible to use PESQ for rating outputs even from a non-bit-exact source like RTIS. However, it is a speech quality measure, not a translational equivalence measure. PESQ analysis was performed on test case 2 to examine if the scores indicate the true perceived quality of the RTIS output.

According to the definition [ITU01], PESQ reference is the original input signal, and the degraded signal to be compared is the result of passing the original input through the network. In RTIS this comparison is not possible, because the signals are essentially different, input English and output Chinese. Therefore, the output of RTIS has to be compared with the desired output, which is a synthesized speech pattern not passed through a network. References defined in test cases above can thus be used. This way we can see how the network affects the signal on its way from RTIS TTS module to the user. Table 8.2 shows the PESQ scores of each reference compared with the recorded RTIS output.

Table 8.2. PESQ analysis results of test case 2.

Reference	**Degraded**	**PESQ Score**
1: 你好你今天好吗?	你今天吗？(recorded)	0,441
2: 你今天好吗?		1,418
3: 你好你今天怎么样?		0,828
4: 你今天怎么样?		0,514
5: 你身体好吗?		0,298
6: 你身体怎么样?		0,028
你今天吗？(synthesized)		4,5

Even though there can be seen a slight correlation between the PESQ score and the number of matching words (see the higher rate given to reference 2), the PESQ scores reflect neither the quality of the speech signal nor the true comprehensibility of the translation. As we can see from the results, a word-exact match is the only case when PESQ analysis result can be considered valid. References 1-6 are not word-exact matches with the recorded RTIS output. This represents extreme temporal clipping which is a situation where PESQ is not intended to be used.

8.2 Analysis of the test environment

The purpose of designing the RTIS automated test environment was to give answers to the following questions:

- Can we adapt Ericsson's speech quality test setup to support RTIS service verification?
- If not, what are the main problems and restrictions? How can we fix the problems and improve the test accuracy?

This chapter discusses the findings from different aspects.

8.2.1 Network effects

One of the main goals of designing this test environment was to show how the network functions, noise, and delay affect the speech transmission when a translation service is involved in a call. As there are no widely used speech-to-speech machine translation services available, no extensive knowledge of the mobile network and mobile environment effects on machine translation performance exists yet.

Network-originated disturbances can be effectively spotted with the help of bit-exact analysis. A bit-exact analysis verification approach in speech quality testing has given us a lot of good experience, which suggests that the same verification philosophy could be adapted to RTIS testing as well. The requirement for performing bit-exact analysis by Upload-BE is that the traffic sent by DUT has to be bit-exact. That is, given a specific input pattern, DUT produces the bit-exactly same speech stream under any circumstance. The result of the example test case 3 in table 8.1 shows that PCM traffic generated by the currently used translation engine is bit-exact and thus can be verified by Upload-BE. As experiments have shown, the output of the translation algorithm is ambiguous, but the TTS module generates the traffic in a bit-exact manner, i.e. the same word sequence always produces the same PCM coded speech stream. However, AMR coded speech behaves differently (see chapter 8.2.4), and this finding is a major deficiency in the RTIS test environment.

When bit-exact analysis is applied in RTIS verification, separating network-originated disturbances from errors caused by a faulty translation is crucial. The closeness measure provided by RTIS test environment does not explicitly indicate if the detected bit-exact errors originate from faulty interpretation of the stimulus in the translation engine, or the network distorting the stimulus. The numeric error measure can help to separate translational mistakes from network errors, but not differentiate the exact source of the error. However, this can be checked with a relatively small amount of tester intervention. The test logs store the location and content of the defected packets; a large number of errors grouped together indicates a faulty translation, whereas network-originated errors occur more randomly. Alternatively, the streams recorded by Lupine and the Linux machine interconnecting M-MGw and the translation engine can be listened to manually to verify if the correct word sequences were sent and received at different points of RTIS execution. This could be automated in the future versions of RTIS (prototype stage 2) test environment, by requesting the translation also in text format and performing a text-based comparison in addition to the speech stream analysis.

8.2.2 Translational equivalence

Another important goal of RTIS verification is to ensure that the service preserves the translational equivalence. Even if the network is able to transmit the signal ideally, a faulty translation should not be accepted. The translations produced by RTIS can vary in their word choices and ordering, so the comparing system needs to be able to respond to this without falsely passing or failing the test case. The multi-SER comparison method used in this test setup allows for some variation in word selection and ordering. The number of references and their structure has to be defined by a bilingual test expert to obtain appropriate test coverage. With the current limited translation capabilities of the system, the coverage obtained by using multiple, fixed references can reasonably be considered adequate. This assumption can be based on manual experiments and the results from the example test run.

As the trial runs show, the test case result does not strongly correlate to human judgement: a human translator would rate all used reference translations correct, whereas the automated test environment rejected several references, rated one reference slightly defected (18%) and two significantly defected (42% and 47%). However, if we only look at the references that didn't get rejected, a correlation between the percentage and the closeness to the actual output can be seen. The highest rated reference deviates from the output by only one word, whereas references 5 and 6 contain less matching words. The percentages also seem to be in a somewhat correct proportion to the true result and to each other. On the other hand, the RTIS test strategy is to find a word-exact match, and when one is found, the number of rejected references does not affect the test result. In a limited-scope translation system it is possible to define reference sets with which the word-exact match can be found for a very high percentage of translations. The poor correlation between multi-SER and human judgement only causes a risk for false rejection, not false acceptance.

Thanks to the flexibility of the tools and test case definition, this test environment can be used when the domain and range of phrases in the translation engine expands: more references can easily be added to the existing test cases, and some of the references can be replaced if needed. Hence, from the standpoint of measuring the translational equivalence, the multi-SER based evaluation method will be feasible in RTIS verification at some point in the future. However, the domain and scope of the translation application needs to remain somehow restricted for several reasons, such as memory issues, test execution time or test case design. In a general-domain MT verification system, there would be practically unlimited number of reference translations per each stimulus. A test expert would have to go through all the references manually when designing a test case to ensure appropriate coverage. A large number of references in each test case would increase the execution time significantly and cause memory issues in storing the reference database. When machine translation development improves, the RTIS setup may need to be replaced or possibly accompanied with a more flexible comparison system based, for example, on BLEU or METEOR.

8.2.3 Tool-related limitations

The tools used in the RTIS test environment bring some restrictions to the verification, as they are not especially designed for MT evaluation. One of the main restrictions is the synchronization required to start Upload-BE analysis. Upload-BE doesn’t synchronize the reference with the incoming stream and thus can’t start the analysis if the incoming stream doesn’t contain a packet that matches the beginning of the reference. In practice this means that if the first word of the incoming stream is faulty, the synchronization might not be accomplished, even if the rest of the reference contains mostly correct words (see results in table 8.1). The synchronization is also lost if a non-matching word is encountered in the middle of the sentence in the incoming speech stream, and thus Upload-BE might consider all the subsequent words faulty as well. Considering our verificational goals, this can be seen as both a good and a bad feature.

From the perspective of verifying the translational equivalence, it is not desirable that a whole sentence can be discarded based on the first word alone (or part of a sentence, if a faulty word occurs later). There is no real risk of false acceptance, but we might falsely reject a large number of references which, according to human judgement, would be rated as understandable. In automated tests with no human supervision, this might give an unrealistic impression of a poor system performance.

On the other hand, we can not be certain that the translated utterance makes any sense even if most of the words match the reference. Only one word might carry the most essential information of the sentence’s meaning, which is lost if that word is missing. Misplaced words might break the syntax and, consequently, the meaning. A 100% position-dependent word match is the only occasion when an automatic MT evaluation system can be certain that a machine translated sentence is correct and comprehensible. Therefore, we should strive to find the exhaustive reference sets, with which the synchronization could be accomplished most times, and the number of false rejects could be minimized. Furthermore, we can only measure and analyze the network effects on translation if the words in the incoming stream match the reference exactly. From this perspective, finding the word-exact match is only the starting point for a successful verification. Therefore, the synchronization required by Upload-BE does not limit the functionality of RTIS test environment.

An issue remarkably increasing the human effort is that the stimuli have to be pre-recorded and stored in the DSP memory before running the test case. Thus, we have to record separate, designated stimuli to be able to test how different speakers, background noise environments, or accents affect the speech recognition.

8.2.4 Compressed speech

To provide adequate end-to-end performance test coverage, the RTIS test environment needs to be able to handle different speech codecs. This possibility was investigated by feeding AMR coded speech directly to the translation engine.

AMR transcoders in a UMTS network can be set to a known state by sending a predefined homing pattern before the actual speech stream to be verified. This way, the output of the transcoder is unambiguous and bit-exact verification is applicable. However, when a translation algorithm is invoked during the call, the bit-exactness of the output can't be guaranteed. The translation algorithms in the 3rd party translation engine seem to affect the AMR coding algorithm in such a way that the effect of sending a homing pattern is canceled.

This finding implies that the RTIS test environment with bit-exact analysis method can not be used as such for testing calls involving compressed speech. However, the testing of different speech codecs in RTIS can not be omitted. Thus, modifications in the test environment and/or designing new tools are required in the future to support AMR and other codecs used in UMTS network. One option is to apply PESQ algorithm on RTIS tests with compressed speech. However, PESQ is known to provide inaccurate predictions when extreme temporal clipping occurs. In RTIS verification this is the case when there is a word mismatch between RTIS output and the reference, as the results in table 8.2 show. If PESQ is applied, the RTIS output should only be compared with a word match reference. This can be ensured by human intervention, or a text-based pre-analysis.

8.2.5 Other findings

As the results presented in chapter 8.1 show, RTIS involves several technically challenging algorithms, all of which can cause unpredictable problems. Due to these problems, analyzing the feasibility of the RTIS test environment is difficult. However, the experimental test case 3 (see table 8.1) proves that it is theoretically possible to apply the concept of automatic bit-exact verification in RTIS testing: a matching word sequence can be found for the translations and bit-exact verification is possible for PCM calls. Improvements in speech activity detection and other speech processing functions involved might open up new possibilities for automated verification and improve the range of RTIS test environment.

The RTIS test environment designed in this thesis can be utilized in regression testing. After a comprehensive test suite has been defined and successfully run on one version of RTIS service, the same test environment and test suite can be used in cyclic regression tests. Regression testing ensures that the existing functionality and features are not broken by the updates in RTIS network environment or the translation engine itself. The RTIS test environment treats the translation engine, which contains the most essential algorithms for the translation, as a black box. This makes running the regression relatively easy and fast even if the whole translation module is replaced with one by a different provider, or its functionality otherwise changes radically.

Considering the functionality level of RTIS prototype and machine translation applications in general, it can be stated that automatic verification is not the most suitable MT evaluation strategy at present. Every stage of the translation process is extremely error-prone: speech recognition, speech activity detection, translation, speech synthesis, and data transmission. Problems occurring on any stage can affect the execution of later stages, which can sometimes multiply the effect of the error. This makes the behavior of the RTIS service very unpredictable, and hence difficult to verify automatically. Together with these findings, the subjective nature of the MT evaluation problem as such suggests that automated verification can't replace manual verification completely in near future. Even covering an extensive part of verification automatically, like in speech quality verification, requires significant improvements in our knowledge of ASR, MT, and MT evaluation methods.

8.3 Future work and open questions

The limited functionality of the current version and the problems found with the speech activity detection make the appropriate verification of the RTIS prototype difficult. Therefore, the test environment designed in this thesis has to be re-evaluated with an improved version of RTIS in the future. However, some specific directions for the future work can already be proposed.

The analysis of the RTIS test environment showed that automatic verification is not necessarily optimal in machine translation evaluation from the translational equivalence point of view. However, some quantitative analysis is needed to ensure that the translation algorithm does not disturb other network functions and that the integration of the service is completed successfully. This suggests that one possible verification strategy in the future could be to separate the translational equivalence evaluation and network analysis. It would require developing a new, partially human-controlled hybrid verification system. It could be partly based on the RTIS test environment designed in this thesis, and partly on a graphic user interface. The test user would be able to monitor the ASR result, forward-translation and back-translation in text format, reject any of the results and send a new request. Only the translation accepted by the test user would be played back in speech format and the network disturbances would be verified. In a hybrid test environment like this, the evaluation would be divided so that a human evaluator is responsible for the translational equivalence rating, and the network behavior is verified by means of e.g. bit-exact analysis. Many of the features in the user interface described above are planned to be implemented in RTIS prototype stage 2, so developing a verification system would also contribute to improving the usability of the actual service.

Verification of compressed speech in an issue that needs to be solved when RTIS IWD is updated to support AMR coded speech. Because we can't verify the network disturbances by bit-exact analysis in AMR calls, another method has to be found. PESQ-based analysis is one considerable option, because the algorithm does not require bit-exactness between RTIS output and the reference. PESQ algorithm rates the utterances automatically, but in the current RTIS test environment realization some manual work is needed to employ PESQ analysis. The test user would have to select the correct reference that matches the RTIS output word-exactly before PESQ comparison can be conducted reliably. Load and stability tests can not be run using PESQ analysis at the moment, because there is no tool running on M-MGw that would set up calls, calculate PESQ scores and monitor the quality over a longer time period and with multiple calls. In the future, different methods for verifying compressed speech should be analyzed and the necessary modifications for automating the execution of the chosen method must be implemented.

A big open question in MT evaluation, that no automatic test environment explicitly answers, is how to define the acceptance level for MT performance. The outputs of RTIS are ambiguous, i.e. the same input sequence can produce variable outputs. More importantly, the translation quality varies between different outputs: some of the translations are "perfect", some have minor style errors or slightly inappropriate word choices, whereas some have unacceptable syntax errors. The longer the input sentence, the more variability there is in the translations. Therefore, in addition to a suitable test method and a working test environment, we need to define a performance level measure. How many test cases are required to prove, for example, that 90% of the translations are rated at least 90% correct? How many test runs are required per test case to ensure that the variability in the quality of the outputs remains within that percentage? What percentage corresponds to human perception of adequate fluency? These questions, however, are not a concern until we can successfully and consistently rate the translated sentences. We need to be able to state with no uncertainty, that a particular sentence meets some quality level (90% correctness, score 4 on Likert scale, etc.) and that the level also corresponds to human judgement. Hence, the ultimate goal in MT evaluation for now is to be able to formulate an accurate evaluation metric.

In speech-to-speech MT verification, a similar question arises regarding the acceptance level of automatic speech recognition. ASR performance has a significant impact on speech-to-speech MT performance. In RTIS test environment, the effect of different speakers, accents, and background noise environments on ASR performance can be verified by performing the test on stimulus files with different characteristics. However, as speech signals are speaker-dependent, and every speaker is unique, we have to find some appropriate generalization for ASR quality. For example, if we perform the test on recorded speech of N different speakers, can we assume that ASR performs adequately for 90% of total end users? Does 90% correctness level in ASR guarantee at least 90% correct translations?

9 Conclusion

This thesis studied speech-to-speech machine translation development and machine translation evaluation. The main goal of the study was to investigate how machine translation works, how the performance of machine translation systems can be evaluated, and how these methods can be applied in verification of a speech-to-speech machine translation service running in a UMTS network. The service to be verified was the Real-time translation system (RTIS), developed by Ericsson.

Machine translation, especially speech-to-speech machine translation, is a very difficult technological challenge. Due to their complexity and ambiguity, natural languages are very difficult to model formally. Implementing accurate speech recognition and natural-like speech synthesis are also complicated tasks, due to the speaker- and context-dependent nature of speech signals. Despite all the challenges, machine translation development has taken leaps forwards in recent years. Statistical machine translation, a relatively new approach, has significantly improved the machine translation performance and encouraged several companies to develop their own machine translation applications. This has resulted in an emerging need to measure and verify the performance of a machine translation application. However, as quality of a translation is a very subjective measure, a truthful, objective evaluation method for MT performance is very hard to define.

Machine translation evaluation methods can be divided into subjective and quantitative methods. In subjective evaluation, a human translator rates the translated sentences using e.g. a Likert scoring system or a task-oriented evaluation model. Subjective evaluation is reliable and reflects the true quality of machine translation, because human translators possess a strong language model and understanding of semantics and the language itself. The main problems with subjective evaluation methods are the differences between the individuals who perform the rating, and also the huge evaluation costs that come from manual, time-consuming work conducted by skilled multilingual experts. Therefore, there is a strong desire to develop quantitative machine translation evaluation methods, with which the evaluation could be performed automatically. The benefits and shortcomings of automatic MT evaluation methods are basically the opposite to those of subjective methods: evaluation costs are low, the quality metric is mathematically formulated and reproducible, but the result does not necessarily correlate to human judgement: correct translations can be falsely rejected and incorrect translations falsely accepted.

The quantitative MT evaluation methods are mostly based on calculating the percentage of correct content. In WER and SER metrics, this is done by comparing the message with a reference translation, and counting the number of correct words or sentences, respectively. These methods are very dependent on the used reference, and different word choices and ordering can skew the result significantly. This can be avoided either by applying some more sophisticated evaluation method, such as BLEU or METEOR, which allow restructuring and combining parts of the reference, or applying multiple references in WER or SER.

The feasibility of applying some quantitative evaluation method in RTIS verification was investigated in this thesis. RTIS is a speech-to-speech machine translation service prototype with limited travel-related scope. The current version of the prototype can only be used in one-phone calls, i.e. the user connects to the translation server with a mobile phone, and his/her translated utterance is played back to the same device. The purpose of developing the prototype was to show how the speech recognition and the actual translation process are affected by the mobile environment and its challenges, such as background and network noise, transcoding, and other speech processing algorithms. Moreover, we want to make sure integrating the translation server into the network doesn't disturb the normal network functionality. The evaluation costs set some requirements for the design of the RTIS test environment: the verification should be automated, and preferably be set up utilizing the existing test tools. The test environment should be able to examine the speech streams on bit stream level, and most importantly, measure how well the translational equivalence is preserved in the mobile environment.

Due to these requirements, the possibility to modify Ericsson's existing speech quality test environment to support RTIS verification was investigated. Both processes involve sending a speech signal into the network and verifying the outcome, but RTIS involves several complex, highly nonlinear algorithms that can change the outcome in various ways.

The first step in RTIS test environment implementation was to reconfigure the setup: the test tools and devices in M-MGw were connected to RTIS interworking device, which routes the generated speech traffic out of the M-MGw, via a Linux server to the external 3rd party translation engine. The translation engine sends the translated utterance back to M-MGw, where it is compared bit-exactly with a pre-defined reference pattern by a special tool called Upload-BE. Defining these references and the logic for rating the translations, i.e. a translational equivalence model, was the major modification required to support RTIS verification.

Because natural languages are ambiguous, there can be multiple correct translations for each input utterance. This is also how RTIS works: the output depends on the translation engine's current state, that is, what preceded the input. The test system needs to adapt to these variations and not classify all word choices and reorderings deviating from the reference as errors. Thus, BLEU/METEOR-based evaluation logic and WER/SER with multiple references were considerable options. The translational equivalence model was chosen to be multi-SER, because in a limited-scope MT system, it can be assumed to provide equal coverage compared to BLEU or METEOR with less initial effort and less complex algorithms. Furthermore, the functionality of Upload-BE was extended to include a closeness measure calculation, which returns the total number of defected bytes to the control interface.

Finally, an experimental test suite was written in TTCN3 to verify the applicability of the RTIS test environment. The test suite consists of two test cases, which were run, and the results analyzed. The following conclusions can be drawn from the results.

As was mentioned earlier, automatic machine translation is a very complex problem. However, speech-to-speech machine translation can be considered exponentially more complex because speech recognition, parsing the spoken language input sentence and speech synthesis are involved in the process. Each of these stages is a very difficult technical challenge, and a minor mistake at any stage can distract the execution of other stages and make the whole translation process fail. Therefore, the difficulties related to verification of speech-to-speech MT applications are not only originating from the inherent complexity and ambiguity of natural languages and the subjective nature of the verification problem itself, but also from the extreme unpredictability of the sub-processes and thus the whole system.

At present, the RTIS service prototype suffers from too many problems and limitations to be verified automatically. The test environment designed in this thesis has to be re-evaluated in the future with an improved version of RTIS prototype, until which manual testing remains the preferred method. Some conclusions can still be stated regarding the applicability of the concept.

Multi-SER as the translational equivalence model can be considered to provide adequate test coverage, as long as the scope of the translation service remains somewhat limited. With a larger scope, the manual work required to define the reference sets and test suite overrides the benefits of automatic verification. Also, issues such as memory consumption and increased execution time become significant. Multi-SER does not correlate strongly with human judgement, but there is only the risk of false rejection, which can be avoided by striving to find an exhaustive reference set for each test case. Because adding test cases, modifying the structure of the test cases and changing the stimulus and reference patterns only requires recompilation of the test suite in RTIS test environment, the multi-SER approach can be considered applicable also in testing the future versions of RTIS.

Upload-BE as a special tool designed to perform bit-exact analysis for the purposes of speech quality testing, is not optimal to be used in RTIS verification. The network disturbances can be spotted very effectively by means of bit-exact analysis, but several findings promote the design of a special RTIS tool: the false rejections caused by the synchronization required by Upload-BE, the fact that AMR (and most probably any other compressed speech) traffic generated by the 3rd party translation engine is not bit-exact, and that the input speech has to be pre-recorded. Research is needed to find the best method for RTIS verification, but the findings of this thesis suggest that text-based analysis and human intervention are needed to some extent. Methods like bit-exact analysis or PESQ solely analyze speech streams and the results can not directly be interpreted as MT quality scores.

One of the major findings of this thesis is that automatic MT evaluation can't replace manual evaluation in near future. Even automating a large share of the verification and running high-load system tests is not likely to be feasible in the next few years. It is possible to build an automated test environment for RTIS and execute some tests successfully, but the true performance level of the application can only be perceived by a human evaluator. Humans' language model, knowledge of the world, understanding of context and the complex discourse analysis capabilities are still astonishing compared to those of the most developed automatic evaluation methods. The test cases in the RTIS test suite show, that even the simplest phrases can be highly ambiguous when they are translated between languages that have significant constitutive differences. Humans can disambiguate and extract the meanings from complex sentences effortlessly, whereas automatic evaluation methods often struggle with basic expressions. The accuracy and truthfulness of the evaluation results vs. evaluation costs still favors manual, human-conducted evaluation. While this holds true, the preferable direction to improve RTIS test environment is to strive to machine-aided human evaluation, that is, reduce the amount of manual work as much as possible, yet still leave the most critical work for humans.

References

[Sta10] D. Stallard, R. Prasad, S. Ananthakrishnan, F. Choi, S. Saleem, P. Natarajan, *Evaluating Different Confirmation Strategies for Speech-to-speech Translation Systems.* Proceedings of IEEE International Conference on Acoustics and Signal Processing, pp. 5218 – 5221, 2010.

[Tom03] J. Tomás, J. À. Mas, F. Casacuberta, *A Quantitative Model for Machine Translation Evaluation*. Proceedings of the EACL 2003 Workshop on Evaluation Initiatives in Natural Language Processing, Apr 2003.

[Alb09] J. S. Albrecht, R. Hwa, G. E. Marai, *Correcting Automatic Translations through Collaborations between MT and Monolingual Target-Language Users*. Proceedings of the 12th Conference of the European Chapter of the ACL, pp. 60-68, Athens, Apr 2009.

[Lop08] A. Lopez, *Statistical machine translation.* ACM Computing Surveys, Vol. 40, No. 3, Article 8, Aug 2008.

[Ros02] T. D. Rossing, F. R. Moore and P. A. Wheeler, *Science of Sound*. 3rd Edition. Addison Wesley, 2002.

[Sha00] D. O'Shaughnessy, *Speech Communications: Human and Machine*. 2nd Edition. IEEE Press, 2000.

[Jur08] D. Jurafsky and J. H. Martin, *Speech and Language Processing.* Pearson Education, 2008.

[Räs07] O. Räsänen, *Speech Segmentation and Clustering Methods for a New Speech Recognition Architecture.* Master's Thesis, Helsinki University of Technology, Laboratory of Acoustics and Audio Signal Processing, Nov 2007.

[Coo07] V. J. Cook, M. Newson, *Chomsky's Universal Grammar: an Introduction.* 3rd Edition. Blackwell Publishing, 2007.

[Iwa00] L. M. Iwanska and S. C. Shapiro, *Natural Language Processing and Knowledge Representation.* AAAI Press, 2000.

[Nir87] S. Nirenburg, *Machine Translation: Theoretical and Methodological issues*. Cambridge University Press, 1987.

[Kit91] N. Kitawaki, K. Itoh, *Pure delay effects on speech quality in telecommunications.* IEEE Journal on Selected Areas in Communications, Vol. 9, Issue 4, pp. 586-593, 1991.

[Jay57] E. T. Jaynes, *Information Theory and Statistical Mechanics.* Physical Review Vol. 106, No. 4, pp. 620–630, May 1957.

[Rai11] T. Raitio, A. Suni, J. Yamagishi, H. Pulakka, J. Nurminen, M. Vainio, P. Alku, *HMM-Based Speech Synthesis Utilizing Glottal Inverse Filtering.* IEEE Transactions on Audio, Speech and Language Processing, Vol. 19, Issue 1. pp. 153 – 165, 2011.

[Tok02] K. Tokuda, H. Zen, A. W. Black, *An HMM-based speech synthesis system applied to English.* Proceedings of 2002 IEEE Workshop on Speech Synthesis, pp. 227 – 230, 2002.

[Tok09] K. Tokuda, H. Zen, A. W. Black, *Statistical Parametric Speech Synthesis.* Speech Communication, Vol. 51, Issue 11, pp. 1039-1064, Nov 2009.

[Tor08] H. M. Torres, J.A. Gurlekian, *Acoustic speech unit segmentation for concatenative synthesis.* Computer Speech & Language, Vol. 22, Issue 2, pp. 196-206, Apr 2008.

[Gee05] D. Geer, *Statistical machine translation gains respect*. Computer, vol. 38, issue 10, pp. 18-21, Oct 2005.

[Eri10a] Oy LM Ericsson Ab, *Real-Time Translation Service: Business Opportunity and Demo*. Aug 2010.

[Cho56] N. Chomsky, *Three Models for the Description of Language.* IRE Transaction on Information Theory, Vol. 2, pp. 113-124, 1956.

[Mit97] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.

[Wan97] Y. Wang, A. Waibel, *Decoding Algorithm in Statistical Machine Translation.* Proceedings of the 8th conference on European chapter of the Association for Computational Linguistics, pp. 366-372, Jul 1997.

[Lev00] L. Levin, D. Gates, A. Lavie, F. Pianesi, D. Wallace, T. Watanabe, M. Woszczyna, *Evaluation of Practical Interlingua for Task-Oriented Dialogue.* Proceedings of the 2000 NAACL-ANLP Workshop on Applied interlinguas: practical applications of interlingual approaches to NLP, Vol. 2, pp. 18-23, Apr 2000.

[Pap02] K. Papineni, S. Roukos, T. Ward, W. J. Zhu, *BLEU: a Method for Automatic Evaluation of Machine Translation*. Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL), pp. 311-318, Philadelphia, Jul 2002.

[Lav07] A. Lavie, A. Agarwal, *METEOR: An Automatic Metric for MT Evaluation with High Levels of Correlation with Human Judgements.* Proceedings of the Second Workshop on Statistical Machine Translation, pp. 228-231, Prague, Jun 2007.

[Por06] M. Porter, *The Porter Stemming Algorithm.* Jan 2006. http://tartarus.org/~martin/PorterStemmer/index.html [Jan 20th 2011]

[Mil10] G. A. Miller, *WordNet: A lexical database for English.* Princeton University, Oct 2010. http://wordnet.princeton.edu/wordnet/ [Jan 20th 2010]

[ITU96] ITU-T Rec. P.800, *Methods for Subjective Determination of Transmission Quality.* Aug 1996. Available http://www.itu.int/rec/T-REC-P.800-199608-I/en [Jan 20th 2011]

[ITU01] ITU-T Rec. P.862, *Perceptual Evaluation of Speech Quality (PESQ): An objective method for end-to-end speech quality assessment of narrow-band telephone networks and speech codecs.* Feb 2001. Available http://www.itu.int/rec/T-REC-P.862-200102-I/en [Jan 20th 2011]

[Sjö11] D. Sjöstrand, Expert interview. Jan 12th, 2011.

[3GP09] 3GPP Technical Specification 23.101 V.9.0.0, *General Universal Mobile Telecommunications System (UMTS) Architecture (Release 9).* 3rd Generation Partnership project, Dec 2009.

[Wit00] A. Witzel, *Control servers in the core network.* Ericsson review No. 4, pp. 234-243, 2000. Available http://www.ericsson.com/ericsson/corpinfo/publications/review/2000_04/files/2000044.pdf [Jan 20th 2011]

[Eri10b] Oy LM Ericsson Ab, *Technical Product Description, M-MGw R6.* Jul 2010.

[Fyr00] M. Fyrö, K. Heikkinen, L. G. Petersen, P. Wiss, *Media Gateway for Mobile Networks.* Ericsson review No. 4, pp. 216-223, 2000. Available http://www.ericsson.com/ericsson/corpinfo/publications/review/2000_04/files/2000042.pdf [Jan 20th 2011]

[Eri09] Oy LM Ericsson Ab, *Real Time Translation Service.* Apr 2009.

[Eri10c] Oy LM Ericsson Ab, *Real Time Interpretation Service (RTIS).* Aug 2010.

[Eri10d] Oy LM Ericsson Ab, *Media Processing Device Load Module Test in R6.1.* Test environment description, Mar 2010.

[Eri06] Oy LM Ericsson Ab, *Upload-BE Quick Study Report.* Dec 2006.

[ETS10] ETSI European Standard 201 873-1, *Testing and Test Control Notation version 3 (TTCN3), part 1: core language.* Version 4.2.1, Jul 2010. Available http://www.ttcn-3.org/StandardSuite.htm [Jan 20th 2011]