Sprinter: Language Technologies for Interactive and Multimedia Language Learning

♯Renlong Ai, ♯Marcela Charfuelan, ♯Walter Kasper, ♯Tina Klüwer, ♯Hans Uszkoreit, ♯Feiyu Xu, ⋆Sandra Gasber, ⋆Philip Gienandt

♯German Research Center for AI
Alt-Moabit 91c, 10559 Berlin, Germany
{*FirstName.LastName*}@dfki.de

⋆LinguaTV
Milastr. 4, 10437 Berlin, Germany
{*FirstName.LastName*}@linguatv.com

Abstract

Modern language learning courses are no longer exclusively based on books or face-to-face lectures. More and more lessons make use of multimedia and personalized learning methods. Many of these are based on e-learning solutions. Learning via the Internet provides 7/24 services that require sizeable human resources. Therefore we witness a growing economic pressure to employ computer-assisted methods for improving language learning in quality, efficiency and scalability. In this paper, we will address three applications of language technologies for language learning: 1) Methods and strategies for pronunciation training in second language learning, e.g., multimodal feedback via visualization of sound features, speech verification and prosody transplantation; 2) Dialogue-based language learning games; 3) Application of parsing and generation technologies to the automatic generation of paraphrases for the semi-automatic production of learning material.

Keywords: multimedia personalized learning, 2nd language learning, pronunciation training

1. Introduction

In the globalized world, the ability to understand and speak the language of business partners and customers or at least to converse with them in English, the lingua franca of business, is rapidly gaining importance in all sectors of the economy. Therefore, the language education market is booming. In recent years, we witness a clear trend towards web-based, interactive, multimedia and personalized learning technologies. Their commercial success does not only depend on the quality of the services but also on the degree of automation. Language technologies can contribute to modern language learning since they provide methods for automating parts of language teaching without giving up on quality. Application areas within language teaching are:

- automatic generation, processing and analysis of learning material
- interactive learning methods
- personalized and individualized learning

In the Sprinter[1] project funded by the German Federal Ministry for Education and Research, three research goals have been pursued for improving automated web-based interactive and multimedia language learning:

- Speech verification for enhancing pronunciation of learners of a second language
- Dialogue technologies for a new type of language learning game
- Paraphrasing for generation of language learning material

For carrying out the research, the Berlin-based company LinguaTV, an online platform for learning languages, focused on producing videos and games for language learning, joined forces with the Language Technology Lab of the German Research Center for Artificial Intelligence (DFKI).

In this paper we will describe how progress in the three lines of research could be achieved by producing and utilizing dedicated specialized language resources. For building the resources efficiently, new tools for data annotation needed to be devised. However, this research is at the same time a strong example of resource reuse. Among the reused resources are treebanks, parsers, grammars, dialogue act inventories and WordNets.

2. Speech Verification

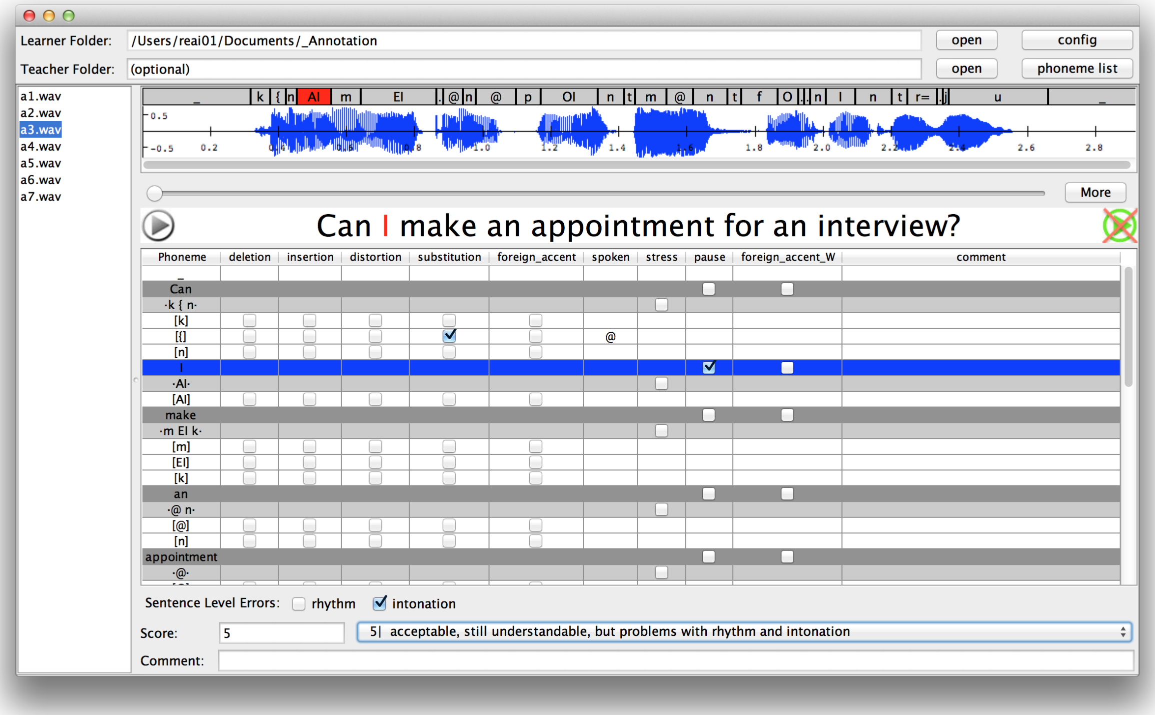
The goal of speech verification in the pronunciation training is to help language learners to recognize and understand their errors via automatic feedback. The automatic feedback has been realized in two ways:

- Feedback via visualization of pronunciation errors and differences between the learner and the native voices
- Audio feedback via speech transplantation, namely, applying the gold-standard native prosody to the learner's voice.

2.1. Annotation Tool for Pronunciation Errors

One of the recent techniques to provide automatic feedback on pronunciation errors of L2 learners is to recognize these by applying trained statistical models. Therefore, we have designed an annotation tool for examining the speech data in a comfortable way and recognizing the errors easily. As depicted in Figure 1, given an input sentence and the visualization of its pronunciation in waveform, the annotator can listen to the whole sentence or select a specific phoneme or

[1] `http://sprinter.dfki.de`



Learner Folder: /Users/reai01/Documents/_Annotation

Teacher Folder: (optional)

a1.wav
a2.wav
a3.wav
a4.wav
a5.wav
a6.wav
a7.wav

Can I make an appointment for an interview?

Phoneme	deletion	insertion	distortion	substitution	foreign_accent	spoken	stress	pause	foreign_accent_W	comment
_										
Can								□	□	
·k { n·							□			
[k]	□	□	□	□	□					
[{]	□	□	□	☑	□	@				
[n]	□	□	□	□	□					
I								☑	□	
·AI·							□			
[AI]	□	□	□	□	□					
make								□	□	
·m EI k·							□			
[m]	□	□	□	□	□					
[EI]	□	□	□	□	□					
[k]	□	□	□	□	□					
an								□	□	
·@ n·							□			
[@]	□	□	□	□	□					
[n]	□	□	□	□	□					
appointment								□	□	
·@·							□			

Sentence Level Errors: □ rhythm ☑ intonation

Score: 5 | 5| acceptable, still understandable, but problems with rhythm and intonation

Comment:

Figure 1: The Error Annotation Tool

any part of the audio signal for playing back. Check boxes at various levels are provided to mark errors by simply tick the check box in the corresponding error column. Comments at any level are also possible. The following error types are considered as relevant: deletion, insertion, distortion, substitution, spoken, stress and pause.

In comparison to other tools (e.g., EasyAlign (Goldman, 2011)), which display alignment and annotation on Praat TextGrid tiers, this annotation tool presents alignment and annotation in separate views. This is possible because alignment and annotation are stored in different files. Other useful features are also implemented:

- zooming the waveform in case there is not enough space to show each phoneme,
- playing the same sentence with native speech if available and
- showing different diagrams such as spectrogram and pitch contours, enabling analysis from various views.

This tool uses the open source EHMM (Black and Lenzo, 2000) to perform force alignment, and MARY TTS (Schröder et al., 2011), a DFKI open-source speech synthesis platform, as text analyser and speech signal processor. Only the audio data and its transcriptions are needed as input. As output, the annotations are stored in XML format, which is consistent with the schema used in MARY TTS. This tool is written in Java and can be deployed on any machine with JRE, and it is also accessible online with embedded Java Applet. More details about this tool can be found in (Ai and Charfuelan, 2014).

In order to test the tool and perform further experiments a corpus was collected. Gold standard English sentences were recorded by a female and a male teacher. A selected set of these sentences (96), were recorded as well by nine female and two male German learners of English. This corpus was segmented per sentences, forced aligned to the text, and currently is being annotated with the tool previously described.

2.2. Automatic Feedback

Feedback via Visualization The Sprinter system can provide a score by automatically comparing a sentence pronounced by a teacher and the learner, in order to give the learner an overall idea of her/his performance. Due to the forced alignment results from EHMM, the error can be exactly located to the phoneme section in the waveform. Various graphs can be generated for the learner to compare the difference visually, e.g. by displaying the pitch contours of the teacher's and learner's voice, or the learner is shown at which section the tone should be lowered or raised as depicted in Figure 2.

Advanced Audio Feedback As pointed out by (Flege, 1995), simple playback of the native and learner's speech cannot help learners to perceive the difference between the sound they produced and the correct target sound. Hence we developed a more advanced type of audio feedback via prosody transplantation.

In prosody transplantation, the prosody features such as pitch and duration are extracted from the native (maybe teacher's) pronunciation and then imposed onto the learner's speech. As a result, the learner can hear a synthesized version of his/her own voice, but with the right

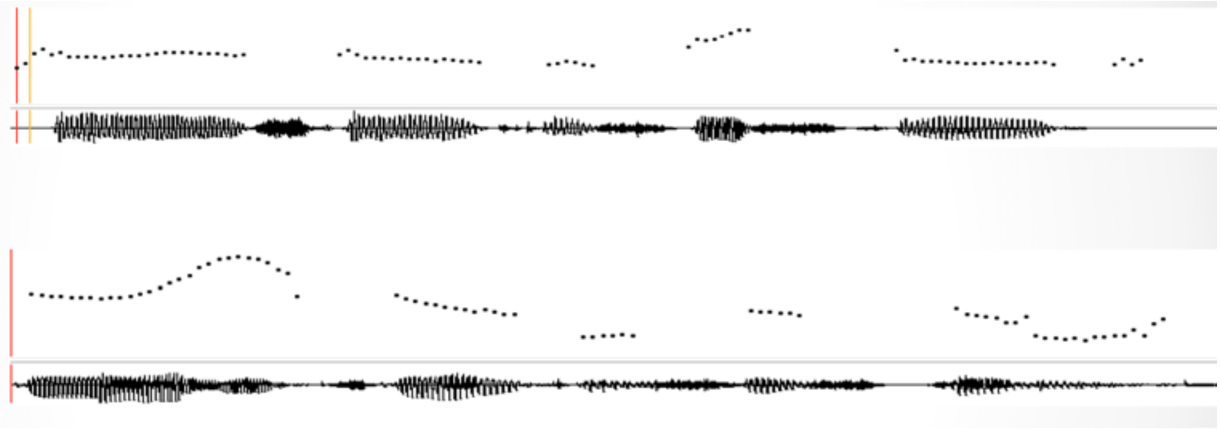


Figure 2: Pitch contours of the voices

prosody. To realize speech transplantation, different methods were tried. Frequency Domain Pitch Synchronous Overlap and Add (FD-PSOLA) (Moulines and Charpentier, 1990), was used for modifying pitch and duration and doing prosody transplantation. Several prosody modification methods have been investigated at DFKI, in particular to perform voice conversion (Turk and Schröder, 2010); in this study it was found that FD-PSOLA, under certain conditions, provides good performance with fewer distortions. Currently we are evaluating the conditions under which prosody transplantation can be provided with minimum distortions. This will be carried out by analyzing the recordings and corresponding annotations. In fact, the detection of different types of pronunciation errors, give us a powerful hint to decide when is safe to provide this type of feedback without distortions, or when it is better to resort to other type of feedback like visual. To generate the transplantation, the pitch and duration features in learner's and teacher's speech data are firstly extracted, using EHMM and Snack Toolkit. After the best alignment of phonemes, the modification parameters are calculated and applied to the learner's speech.

The advantage of prosody transplantation is that it provides not only the corrected prosody, which can be perceived by the learner through comparing the difference between the transplanted speech and his own, but also corrective information of where and how the right prosody should be used. It has been shown that second language learners can imitate more easily, when the target pronunciation is perceived in their own voice (Felps et al., 2009).

3. Paraphrasing

One goal of Sprinter is the use of linguistic tools and resources to support the development of grammar and dialog exercises. Especially desirable is a higher degree of flexibility with respect to possible alternative solutions in addition to a model solution. In the context of the Sprinter project this work is subsumed under the heading of "paraphrasing" as the task of finding possible variants of example sentences that might provide alternative solutions.

As a first application scenario, so-called "Jumbled order exercises" were selected: the learner is presented a number of words or word groups in random order. The task consists of forming from these a grammatically correct sentence. Without an idea of the meaning of the target sentence, this task can be difficult. In today's programs, only a single solution is defined as the correct one even if there might be several possible solutions, which could lead to frustration on the learner's side. We developed a paraphrasing tool for the exercise developers that generates admissible alternative solutions to the task.

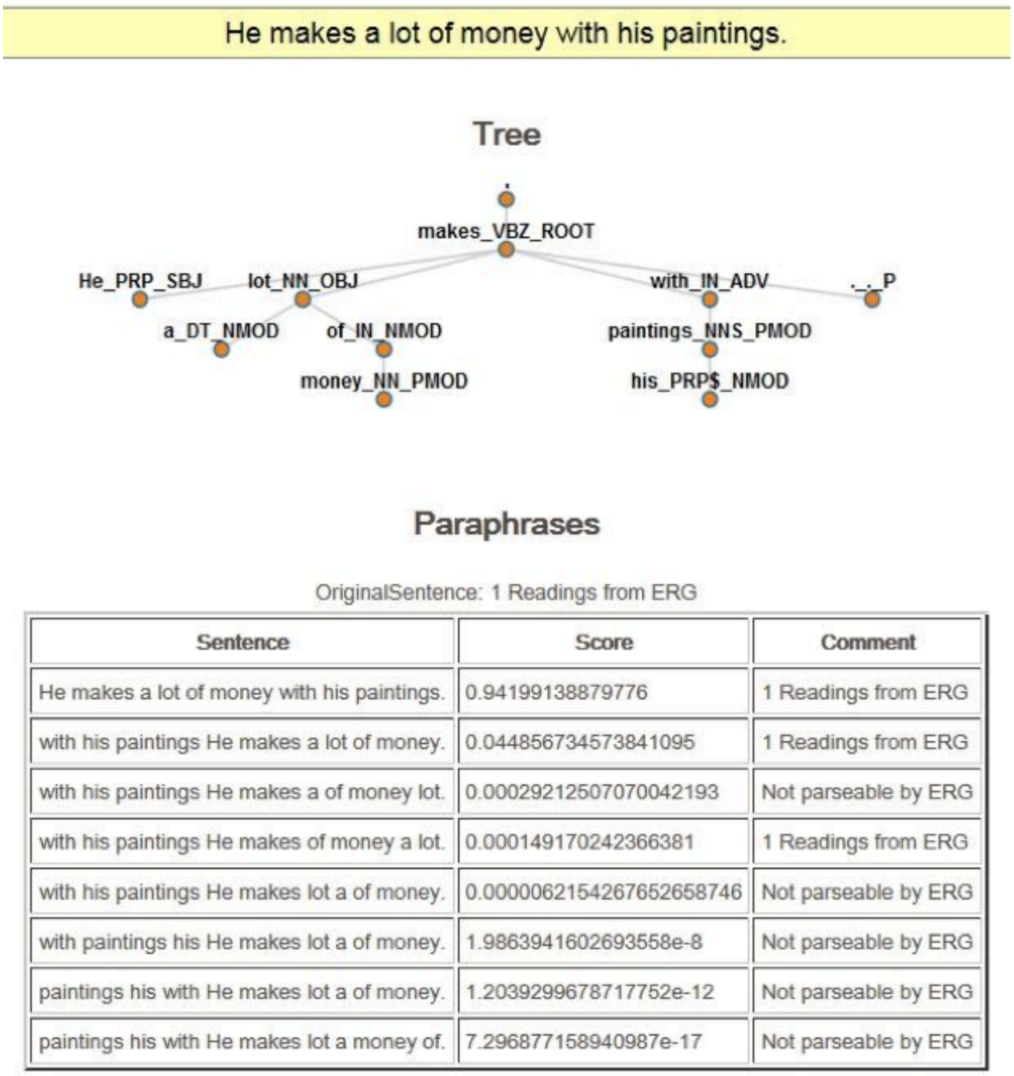
We adopted a "parse and generate" approach that is based on dependency analysis and subsequent generation of sentence variants as different linearizations of the dependency structures. Dependency structures provide a functional representation of a sentence, in general without implying a specific word order, in contrast to phrase structure representations.

For the dependency analysis we employed DFKI dependency parsers trained on dependency treebanks (Volokh and Neumann, 2012). The resulting dependency tree is fed into a generator (Zhang and Wang, 2012) that produces possible linearizations for the dependency tree as sentence variants. To reduce over-generation of ungrammatical variants, in addition a deep parser based on HPSG grammars is used for grammatical verification of the generated sentences. Figure 3 gives an impression of what the analysis and the outcome of the paraphraser system look like.

Advantages of this approach are:

- The number of generated variants can be controlled by various parameters, such as beam size and probability thresholds.
- The design is modular in allowing to combine various kinds of resources and language models for refinements.
- Robustness is achieved with respect to lexicon and syntactic structure.
- User feedback can be used to improve and further adapt the models by re-training easily without requiring specialized linguistic or system knowledge.

User feedback is collected through an Exercise Manager as web-frontend to the paraphraser. The manager allows exercise developers to create, edit and augment exercises



OriginalSentence: 1 Readings from ERG

Sentence	Score	Comment
He makes a lot of money with his paintings.	0.94199138879776	1 Readings from ERG
with his paintings He makes a lot of money.	0.044856734573841095	1 Readings from ERG
with his paintings He makes a of money lot.	0.00029212507070042193	Not parseable by ERG
with his paintings He makes of money a lot.	0.000149170242366381	1 Readings from ERG
with his paintings He makes lot a of money.	0.0000062154267652658746	Not parseable by ERG
with paintings his He makes lot a of money.	1.9863941602693558e-8	Not parseable by ERG
paintings his with He makes lot a of money.	1.2039299678717752e-12	Not parseable by ERG
paintings his with He makes lot a money of.	7.296877158940987e-17	Not parseable by ERG

Figure 3: Web-based paraphrasing tool

with results from the paraphraser. Feedback for the paraphraser's suggestions can be provided as a rating on a 3-level scale as being good, acceptable (in special contexts) or just bad.

Currently, English and German are supported. The dependency parser and generator were trained on dependency versions of the Penn Treebank and the TIGER Treebank, respectively. The deep parsers for grammatical verification are based on HPSG grammars, the English Resource Grammar (ERG (Copestake and Flickinger, 2000)) and the German Grammar (GG (Müller and Kasper, 2000)), both in their actual versions from DELPH-IN[2].

The system was evaluated with the data from 44 English exercises. On average the system generated 3.8 paraphrases for these sentences. In nearly all cases, the original sentence was also reproduced as "best" linearization. We take this as an indication of high reliability of the analysis as well as of the sentence generation. Ungrammatical paraphrases were correctly identified by the English HPSG grammar in 63% of cases.

For the future, the inclusion of tree re-writing methods and lexical resources is planned to extend the system to other paraphrase types, such as lexical paraphrases, diathesis and dialog ellipsis.

4. Dialogue for Language Learning

Sprinter offers an interactive dialogue training in a game-like exercise. In this exercise, users can train their conversation abilities in special prototypical scenarios such as hotel room reservation or business phone calls through text-based or spoken chat with a software agent. In the interactive dialogue training, the users moreover can practise their language skills in general.

For the dialogue training, the language learning platform is connected to a dialogue system that controls the dialogue agents. The architecture of the integration follows a server-client approach. The dialogue agents are controlled by an external dialogue server, while the game itself runs on the client machines. Because a natural dialogue needs to enable mixed-initiative behavior, the communication between the two platforms is asynchronous and realized using the websocket standard. The server is implemented using the

[2] http://moin.delph-in.net/

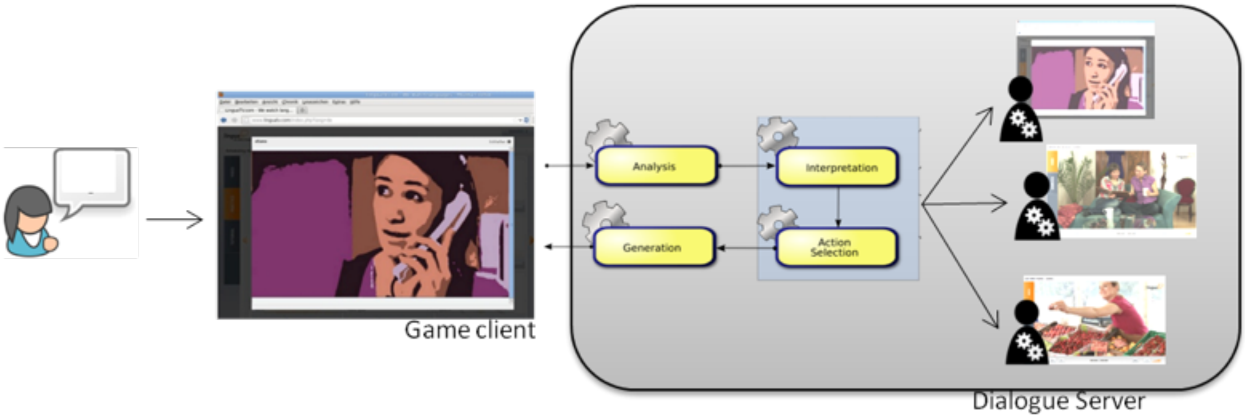


Figure 4: Dialogue-Based Interactive Language Learning

Java 7 Websocket API.

The dialogue agent architecture is independent of the actual scenario. Possible types of agents are the possible game scenarios such as “hotel room reservation” and “appointment phone calls”. The dialogue server manages the classes of agents and the instantiated agent-objects. In the final system for every game started a new agent representing the selected scenario will be created. The agents all use the same dialogue system’s methods for analysis and generation, but differ in their dialogue models, which are scenario-specific. A dialogue model encodes the knowledge about a conversation for a specific scenario.

Dialogue models are learned from conversation data successively. Conversation data originates from the scripts of the LinguaTV videos and additional sources if available. For the current version of the system prototype, data from a Business English video lesson dealing with an appointment phone call was prepared and annotated with dialogue act, topic and sequence information. Additionally, data from the Verbmobil corpus was used. The final data set consists of 120 conversations.

The employed inventory of dialogue acts is a subset of the DIT++ set (Bunt, 2006; Bunt, 2011; Geertzen, 2009) augmented by an additional set for small-talk dialogue acts (Klüwer, 2011). Topics are annotated using a specific ontology for the appointment domain plus WordNet. Topics are URIs of the RDF versions of the knowledge bases.

From the annotated dialogue data, state graphs are learned that represent the dialogue models. Using standard methods for finite state automata, they are then integrated into the final dialogue system.

Figure 4 shows the communication pipeline of the game exercise. A user gives input to the game client. The game client delivers the input to the dialogue server via a WebSocket connection. The server analyzes the incoming data, identifies or creates the dialogue agent for the scenario and retrieves an abstract answer using the agent’s dialogue model. Lastly, the dialogue system generates a response, which is sent back to the game client and there presented to the user.

5. Conclusion

We have presented our research results, which will contribute to interactive and individualized language learning and to automatic generation of high quality learning content.

- Our annotation tool for pronunciation errors provides a convenient way for identifying and annotating the error types of each corresponding phoneme via visualization of sound and audio playback
- The annotated resources enable a combination of visualization feedback and advanced audio feedback via speech transplantation, which helps users to understand their errors and the ways to correct them
- The parse and generate method for paraphrasing heavily reuses existing resources (treebanks, grammars and parsers) in a combination of parsing and generation technology in a novel way for the automatic creation of learning content
- The dialogue-based game provides an interactive situation based learning context where learners can be tested with various learning goals such as pragmatics, word and grammar. To this end it uses annotated dialogue data from video transcripts and existing resources such as dialogue-act inventories and WordNets.

By the number, variety and creative use of existing and new data and tools that were needed to realize novel features of a single application, the described research demonstrates the importance of language resources for product innovation.

6. Acknowledgements

This research was partially supported by the German Federal Ministry of Education and Research (BMBF) through the project Sprinter (contract 01IS12006B) and the project Deependance (contract 01IW11003).

7. References

Ai, Renlong and Charfuelan, Marcela. (2014). MAT: a tool for l2 pronunciation errors annotation. In *Proc. of LREC*, Reykjavik, Iceland.

Black, Alan and Lenzo, Kevin. (2000). Building voices in the festival speech synthesis system.

Bunt, Harry. (2006). Dimensions in dialogue act annotation. In *Proc. of LREC*, volume 6, pages 919–924, Genoa, Italy.

Bunt, Harry. (2011). Multifunctionality in dialogue. *Computer Speech & Language*, 25(2):222–245.

Copestake, Ann and Flickinger, Dan. (2000). An open source grammar development environment and broad-coverage english grammar using HPSG. In *Proc. of LREC*, Athens, Greece.

Felps, Daniel, Bortfeld, Heather, and Gutierrez-Osuna, Ricardo. (2009). Foreign accent conversion in computer assisted pronunciation training. *Speech communication*, 51(10):920–932.

Flege, James E. (1995). Second language speech learning: Theory, findings, and problems. *Speech perception and linguistic experience: Issues in cross-language research*, pages 233–277.

Geertzen, J. (2009). *The automatic recognition and prediction of dialogue acts*. Ph.D. thesis, PhD Thesis, Tilburg University.

Goldman, Jean Philippe. (2011). Easyalign: An automatic phonetic alignment tool under Praat. In *Interspeech*, pages 3233–3236.

Klüwer, Tina. (2011). i like your shirt-dialogue acts for enabling social talk in conversational agents. In *Intelligent Virtual Agents*, pages 14–27. Springer.

Moulines, Eric and Charpentier, Francis. (1990). Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones. *Speech communication*, 9(5):453–467.

Müller, Stefan and Kasper, Walter. (2000). Hpsg analysis of german. In *Verbmobil: Foundations of Speech-to-Speech Translation*, pages 238–253. Springer.

Schröder, Marc, Charfuelan, Marcela, Pammi, Sathish, Steiner, Ingmar, et al. (2011). Open source voice creation toolkit for the mary tts platform. In *12th Annual Conference of the International Speech Communication Association-Interspeech 2011*, pages 3253–3256.

Turk, Oytun and Schröder, Marc. (2010). Evaluation of expressive speech synthesis with voice conversion and copy resynthesis techniques. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(5):965–973.

Volokh, Alexander and Neumann, Günter. (2012). Dependency parsing with efficient feature extraction. In *KI 2012: Advances in Artificial Intelligence*, pages 253–256. Springer.

Zhang, Yi and Wang, Rui. (2012). Sentence realization with unlexicalized tree linearization grammars. In *Proceedings of the 24th International Conference on Computational Linguistics (COLING 2012)*.