Abstract

This paper describes the dynamic, multi-lingual lexicon that was developed in the SmartKom project. SmartKom is a multi-modal dialogue system that is supposed to assist the user in many applications which are characterised by their highly dynamic contents. Because of this dynamic nature various modules of the dialogue ranging from speech recognition over analysis to synthesis need to have one common knowledge source that takes care of the dynamic vocabularies that need to be processed. This central knowledge source is the lexicon. It is able to dynamically add and remove new words and generate the pronunciations for these words. We also describe the class-based language model (LM) that is used in SmartKom and that is closely coupled with the lexicon. Also evaluation results for this LM are given. Furthermore we describe our approach to dynamically generate pronunciations and give experimental results for the different classifiers we trained for this task.

1. Introduction

SmartKom [1] is a German multi-modal dialogue system that combines spontaneous speech and gesture recognition, as well as the recognition of the user’s emotion from his facial expression. A multi-modal output is realised by a life-like character, cf. [2]. SmartKom was developed having three scenarios in mind:

- **SmartKom Home**: This scenario offers a common interface to devices that the user might want to control in his home, such as video cassette recorder (VCR), television (TV), electronic program guide (EPG) and HiFi devices.
- **SmartKom Mobile**: The system is PDA-based and offers car as well as pedestrian navigation. This includes information services about the city at the current user’s location. This scenario is currently also available in English.
- **SmartKom Public**: The system serves as a public information system that e.g. provides hotel information, cinema information, telephony services, etc.

It is obvious that in all of these different scenarios, the content that is to be controlled by the user is highly variable. The TV program/EPG that the user might want to consult when he/she is at home changes daily or even more frequently, navigation in cities requires knowledge of current street names and points of interest of this city and also the cinema program and other public information changes rapidly.

Nowadays systems usually cannot automatically handle this kind of dynamic content. As a result they need to keep as many words as possible in the system dictionary right from the beginning, which increases the dictionary size and thus the speech recognition error rates and also causes the subsequent components (e.g. speech analysis) to become more complex. Furthermore it is often not possible to determine the necessary words in advance (e.g. TV program). Another difficulty is that the dynamic information needs to be handled appropriately by various modules of such a system. For example the speech recognition and prosody recognition modules need to be able to recognise new words, the speech analysis needs to be able to interpret a user’s input possibly referring to the new content and last but not least the speech synthesis needs to be able to generate output that might contain new words. Furthermore it is absolutely necessary that all these modules use the same information to show consistent system behaviour.

For these reasons the lexicon plays a central role in SmartKom. It acts as the central knowledge source for all modules that need to access lexicon information. The lexicon has the ability to dynamically update if new words are to be handled during a dialogue turn. This requires the automatic generation of pronunciations for new words, so that the recogniser can recognise them, the synthesis can synthesize them etc.

This paper is organised as follows: Section 2 explains the dynamic lexicon and its integration in the overall SmartKom system. Section 3 describes the class-based LM that is used in SmartKom and that also needs to reflect the dynamic content. In Section 4 the procedure for automatically generating pronunciations for new words is described. This includes the generation of multi-lingual pronunciations. Section 5 finally summarises the results.

2. The Dynamic Lexicon in SmartKom

In order to explain how the dynamic lexicon is integrated into the SmartKom system, first its overall architecture is shown in Figure 1. This Figure does not show the SmartKom system in detail but focuses on those components that are relevant for describing the functionality of the lexicon. The normal arrows indicate the direct links between the modules, the bold arrows are meant to show in particular the modules that depend on the lexicon. After a user utterance, first the speech input is recognised and a word-graph is output by the speech recognition module. This is taken by the speech analysis component together with the result from the prosody analysis to compute several possible semantic hypotheses. These are then combined with the gesture recognition result in the media fusion. According to the user’s wish the action planning component initiates the necessary actions. In case the user wanted to know which movies are on TV tonight the function model translates this user wish into a database query, where the TV program is kept. The results
of this query are sent back to the action planner. The action planner then initiates a lexicon update by sending the words contained in the search result (in this case movie titles) to the lexicon. The lexicon checks whether these words are already contained in the lexicon and if not adds them and generates the pronunciations for these words. Finally, it informs the speech recognition, prosody and speech analysis about this update.

The communication between modules is established using a pool architecture, where all the data is written to a pool, from where ‘interested’ modules can get the data. As an interface language an XML-based language called ‘XML’ (Multi-Modal Mark-up Language) was developed in SmartKom. For the lexicon update a general pool is used, that any other interested module can ‘listen to’ and read its content. This way the content of the lexicon update needs to be written to the pool only once and is available for all other modules.

If words that are contained in the update request from the action planner are already in the lexicon, typically no pronunciations need to be generated and there is also no need for the subsequent modules to update. However, there are many cases in which a word is already in the lexicon, but has a different word class assigned than the word that is contained in the update request. For example, the German word ‘Königstrasse’ could be a street name (which is already in the lexicon) as well as the name of a cinema (which is contained in the update request). In this case, although no new pronunciation needs to be generated, the word class needs to be added, which means also the other modules need to update, otherwise the speech recogniser could not recognise Königstrasse as a cinema but just as a street name, which would lead to misrecognitions in user requests for certain cinemas. The same holds for examples where a word is contained in the lexicon with a German pronunciation and now the same word is to be added as a movie title. Here, the additional generation of the English pronunciation is necessary (see Section 4.2) as well as the update of the subsequent modules.

After the lexicon was updated and all necessary other steps have been taken to fulfil the user’s request, an appropriate system response needs to be generated. This mainly consists of a graphical representation of the search results, as well as of the animation (e.g. pointing gestures, lip movements) of the avatar together with the words that he is going to speak. So the speech generator generates the system prompt to the user input. Usually this would directly serve as input for the speech synthesis. In the case of SmartKom, the generator sends the generated sentence to a request pool of the lexicon. The lexicon generates the pronunciations for these words and then sends the sentence(s) together with the pronunciation, stress and syllable boundary markers to the response pool from where the speech synthesis can retrieve it and use it to synthesise the system utterance.

So far, we only described the procedure how to add words to the lexicon but there are also a lot of cases where it makes sense to remove words from the lexicon, e.g. in the mobile scenario. If the user requests the navigation within a target city, the corresponding street names are loaded to the lexicon. As soon as this navigation task has been finished and the user switches topic to another task the several thousand street names can be removed again. So the size of the dictionary can be drastically decreased which is beneficial for the speech recognition accuracy.

Of course the content of the lexicon is not completely dynamic. A domain specific vocabulary was defined based on those words that occur in the three scenarios described above and that cover ‘general’ dialogues. This set of 5292 words is contained in the baseline lexicon, that means these words will not be changed. Their pronunciations were manually corrected and they are marked as ‘manual’. Words that are automatically generated by the automatic grapheme-to-phoneme conversion (g2p) are marked ‘automatic’. Furthermore the language, the part-of-speech tag as used by the speech analysis component and the LM class (if any) are contained in the lexicon for each entry.

3. Class-based Language Models

The LM used in this project is a statistical, class-based, 3-gram LM. The LM is tightly coupled to the lexicon and also has to reflect the dynamic nature of the vocabulary. To be able to do so, frequently changing words, such as actor names, movie titles, streets in different cities, etc., are categorised and put into word classes. The 3-grams are then based on these classes, rather than on the words themselves. The advantage of the class-based LM is that it is not necessary any more to have all words in all contexts in the training data. For all words in one class the histories of all the single words in that class can automatically be associated with all words in that class. To give an example: Consider the two sentences: ‘I liked James Bond very much’ and ‘I hated Star Trek’. If a MOVIETITLE class was used, you had ‘I liked MOVIE TITLES very much’ and ‘I hated MOVIE TITLES’. The resulting LM would then cover the original sentences as well as the possible combinations ‘I liked Star Trek very much’ etc. since both ‘Star Trek’ and ‘James Bond’ are in the MOVIE TITLES class.

In SmartKom the content of the word classes can dynamically change. New words can be added/removed to/from a class. We just need to recompute the in-class probability for all words in the classes, which is trivial if all words in a class are equally weighted. However, it would be possible to increase the weight of those words in a class that are currently displayed on the screen. This would account for the fact, that e.g. a list of 20 movie titles is the result of a user request and only 10 can be displayed on the screen (to view the others you would have to scroll down), then the user is more likely to say one of those titles that he can currently see on the screen and therefore their probability could be increased.

By updating the contents of the classes, we can achieve the adaptation of the LM to a specific domain to a certain extent, without the need to completely change and reload the LM (like in VoiceXML) which would be very time-consuming and often not feasible in dialogue systems of this coverage. Furthermore, by not restricting the LM to sub-parts for certain kinds of expected dialogue turns (which is an often used strategy), the ca-

![SmartKom architecture](figure1.png)

**Figure 1:** SmartKom architecture (part of the system)
pability of being able to speak words from all sub-domains, no matter if they are within the range of expected utterances at this stage of the dialogue, remains.

Currently 31 word classes exist. However, not all can be dynamically updated. Those that can be are the following: ACTOR, CITY, GARAGE, GENRE, HOTEL, LOCATION, MOVIE THEATER and MOVIETITLE. The LM is trained using newspaper text (Süddeutsche Zeitung), domain-specific texts from newsgroups and collected dialog turns. After an appropriate pre-processing of the different text corpora, we yield 3297970 sentences for training. The Süddeutsche corpus mainly serves as a background corpus to capture the general characteristics of the German language. The dialogue turns that partly result from the Wizard-of-Oz (WoZ) experiments that are conducted by the LMU Munich [3] are supposed to reflect the specific scenarios for which SmartKom is intended. Also SmartKom is supposed to handle more spontaneous speech, which is not at all reflected in newspaper texts. The collection of newsgroup texts in addition to the dialogue data is necessary, since the amount of dialogue specific data is not sufficient to construct a reliable LM. Furthermore the nature of writing in newsgroups is somehow close to spontaneous speech. We evaluated the LM in terms of test set perplexity. The results are shown in Table 1. We used 9/10 of the sentences for training and 1/10 for evaluation. The size of the vocabulary was 2983 words including classes. We used the CMU toolkit (v2) [4] to train the LM and we trained one LM that covers all scenarios, however, the evaluation was done scenario-wise, since the system will be tested later scenario-wise only, too.

‘mob’ are those dialogues from the mobile scenario (integrated route planning, map manipulation, points of interest, etc.). ‘pub’ are dialogues from the public scenario (telephony, address book, email, fax, cinema, calendar) and ‘home’ are dialogues from the home scenario (device control, EPG, TV, videos). The ‘general’ dialogues cover those dialogues that do not strictly belong to one of the scenarios. You can see that the perplexities vary from 7.7 to 35.5 with the ‘general’ test set having the highest perplexity. We also evaluated the system in terms of word error rate (WER) using the SmartKom recogniser that was developed by DaimlerChrysler. We did so by using different speech corpora that were recorded by different partners. The results are listed in Table 2. ‘pub’ again reflects the public scenario (recorded by DaimlerChrysler) and ‘mobI’ are dialogues from the mobile scenario (pedestrian navigation, recorded by DaimlerChrysler), ‘mobII’ are also dialogues from the mobile scenario (pedestrian navigation, recorded by EML), ‘MTI’ are dialogues for the home scenario that were recorded during the MTI exhibiton in October 2001, SKI and SKII were dialogues recorded in the SmartKom WoZ recordings. The results for the WoZ recordings are rather bad. The reason is that these recordings include huge amounts of off-talk that was completely out of domain (e.g. users starting to read movie descriptions that are displayed in the home scenario when the task was to select a movie, etc.). For the more scenario-specific dialogues that did not include that much off-talk, the results are much better ranging from 8.8% WER to 29.9%. Remember that these experiments included the dynamic update of the word classes and the automatic generation of pronunciations for these words. Please also note that SmartKom is a multi-modal system. Thus by combining speech with gestures a lot of sentences can be interpreted correctly, even if the speech recognition rate is not perfect.

### 4. Automatic Grapheme-to-Phoneme Conversion

For the words that are added to the lexicon, the pronunciations need to be generated automatically. A common approach is to use g2p tools. We use an automatically trained classifier which in our case is a decision tree\(^1\), that was induced from a large set of language-specific data. This means we use existing dictionaries that contain the orthography and pronunciations of the words together with the syllable boundaries and word stress markers. The tree predicts grapheme by grapheme the corresponding phoneme considering the left and right context of the grapheme. This approach is described in more detail in [6].

For SmartKom we have to be able to generate German and English pronunciations, so we used a German and an English training set, respectively, to train the respective trees. During training we tried different parameter settings, that influence the generalisation capabilities of the resulting trees. Also we used different sizes of training sets resulting in differently sized trees to see how much this influences the phoneme prediction accuracies. Furthermore we applied a technique called ‘boosting’. Boosting is supposed to increase the accuracies by training several classifiers instead of one, each classifier focusing on those parts of the training data, that were misclassified by the previous classifier.

#### 4.1. Experiments and Results

We wanted to see the relation between size (measured in number of nodes the tree consists of) and word accuracy rate. We compare the number of classifiers used for boosting (1, 5, 10 and 15) and the sizes of the training data size (20k and 100k). Furthermore we experimented with the pre-pruning parameter which specifies the minimum number of occurrences of cases that must be covered by each branch of the tree. Choosing a small pre-pruning factor thus yields highly specialised trees that will have a large number of nodes. Correspondingly, a large pre-pruning factor will result in smaller trees with less nodes, but stronger generalisation capabilities. We varied the pre-pruning factor from 2 to 256. For the training set we chose subsets of the German and the British English CELEX lexicon [7].

The results for the German experiments can be seen in Figure 2. The different pre-pruning factors are reflected by the varying number of tree nodes. The results do not take into account syllable boundary and word stress error.

The best results in terms of word accuracy can be achieved

<table>
<thead>
<tr>
<th>corpus</th>
<th>pub</th>
<th>mobI</th>
<th>mobII</th>
<th>MTI</th>
<th>SKI</th>
<th>SKII</th>
</tr>
</thead>
<tbody>
<tr>
<td># sent</td>
<td>50</td>
<td>50</td>
<td>50</td>
<td>114</td>
<td>50</td>
<td>50</td>
</tr>
<tr>
<td>perplexity</td>
<td>7.2</td>
<td>14.6</td>
<td>12.6</td>
<td>16.7</td>
<td>19.7</td>
<td>42.2</td>
</tr>
<tr>
<td>WER</td>
<td>10.7</td>
<td>8.8</td>
<td>27.6</td>
<td>25.5</td>
<td>29.9</td>
<td>65.4</td>
</tr>
</tbody>
</table>

1we are using the tool C5.0 [5]
when a large training set and boosting are chosen. Boosting is beneficial but at the cost of very large trees. Smaller trees can be achieved by three factors:

- by reducing the number of classifiers for boosting (indicated by different symbol shapes in Fig. 2)
- by using a smaller training set (white symbols vs. grey symbols)
- by restricting tree growth by means of the pre-pruning parameter (series of identical symbols vary only in the amount of pruning).

Indeed, the smallest trees were generated from a small training set with strong pruning and no boosting (series of white diamond symbols).

Focusing on relatively small trees with up to approximately 1000 nodes, it can be observed that trees trained on the small training set with weaker pruning are comparable in size to trees originating from the bigger set with stronger pruning, with notably better accuracy rates for the former. We thus draw the conclusion that for single classifiers without boosting, the enforced stronger generalisation resulting from a high pre-pruning factor is more harmful than a small training set.

Regarding the larger trees, results are clearly better for the larger training set. Also, the more classifiers are used for boosting, the better the results. However, the accuracies for 10 and 15 classifiers respectively differ only slightly (cf. the grey squares and circles). Moreover, letting the trees grow too much (i.e., applying too little pre-pruning) can be unfavourable or at least does not increase word accuracy any further (cf. the downward trend at the right end of most series). For the very large trees, the critical threshold was about 32 cases per branch. The British English results show the same tendencies (not shown in the figure), except that the accuracies are around 8% lower compared to the German ones.

For the integration into the SmartKom demonstrator we chose a middle sized tree yielding around 85% word accuracy for practical reasons, since larger and thus slightly more accurate trees resulted in very long startup times of the lexicon module. Remember that two of those trees (English and German) are used.

4.2. Generating Multi-lingual Pronunciations

Although SmartKom when used in German mainly works with German pronunciations, it has to handle a lot of non-German content, e.g. music track titles, movie titles, actors’ names etc. These are often English and thus an English pronunciation for these words is required. Currently British English and German pronunciations can be generated in SmartKom using one decision tree for each language. For the speech recognition and synthesis part we mapped English phonemes to the closest German ones where possible. Those phonemes that cannot be mapped are added to the German phoneme set. A description of the used phoneme set can be found in [8]. As the default language is German, German pronunciations are generated (unless requested explicitly otherwise), except for words that belong to the classes ‘ACTOR’ and ‘MOVIE TITLE’. For words in these classes also the English pronunciation is generated, since many of these words are English. The motivation behind this is that the titles do not have language tags attached in the database and there is also no language identification available. Furthermore German speakers often do not use the correct English pronunciation but a rather German one. Thus including both pronunciation often helps to improve the speech recognition rates.

5. Summary

This paper presented the dynamic, multi-lingual lexicon that is used as a central knowledge source for the speech and prosody recognition modules as well as for the speech analysis and synthesis modules in the SmartKom project. It can dynamically add and remove new words and generate pronunciations for these words in German and in English. In this way dynamic contents can be handled flexibly by the dialogue system. We also described the class-based language model that is closely coupled with the lexicon and showed our evaluation results in terms of perplexity and word error rate. Furthermore we described the automatic generation of pronunciations in detail and gave some experimental results for this, reflecting the accuracy of the pronunciation generation.

6. Acknowledgements

This research was conducted within the SMARTKOM project and partly funded by the German Federal Ministry of Education and Research (BMBF) under grant 01IL90517. Thanks to Antje Schweitzer for her help in conducting the g2p experiments.

7. References