Language models beyond word strings

Elmar Nöth, Anton Batliner, Heinrich Niemann, Georg Stemmer, Florian Gallwitz, Jörg Spilker

Angaben zur Veröffentlichung / Publication details:

Nöth, Elmar, Anton Batliner, Heinrich Niemann, Georg Stemmer, Florian Gallwitz, and Jörg Spilker. 2001. "Language models beyond word strings." In *Workshop on Automatic Speech Recognition and Understanding, ASRU'01, 9-13 December 2001, Madonna di Campiglio, Italy*, 167–76. Piscataway, N.J: IEEE. https://doi.org/10.1109/ASRU.2001.1034614.

From Proceedings of the Automatic Speech Recognition and Understanding Workshop (ASRU'01) (to appear)

LANGUAGE MODELS BEYOND WORD STRINGS

Elmar Nöth [*], *Anton Batliner, Heinrich Niemann, Georg Stemmer, Florian Gallwitz*[†], *Jörg Spilker*[‡]

Universität Erlangen–Nürnberg
Lehrstuhl für Mustererkennung (Informatik 5)
Martensstr. 3, 91058 Erlangen, Germany
noeth@informatik.uni-erlangen.de

ABSTRACT

In this paper we want to show how n–gram language models can be used to provide additional information in automatic speech understanding systems beyond the pure word chain. This becomes important in the context of conversational dialogue systems that have to recognize and interpret spontaneous speech. We show how n–grams can (1) help to classify prosodic events like boundaries and accents, (2) be extended to directly provide boundary information in the speech recognition phase, (3) help to process speech repairs, and (4) detect and semantically classify out–of–vocabulary words. The approaches can work on the best word chain or a word hypotheses graph. Examples and experimental results are provided from our own research within the EVAR information retrieval and the VERBMOBIL speech–to–speech translation system.

1. INTRODUCTION

In this paper we want to show how stochastic n–gram language models can be used to provide additional information in automatic speech understanding (ASU) systems beyond the pure word chain. The best word chain $\boldsymbol{w}^*$ in practically all speech recognizers is the result of the fundamental formula of speech recognition, the Bayes' Formula

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmax}} \{P(\boldsymbol{O}|\boldsymbol{w}) \cdot P(\boldsymbol{w})\}$$

where $\boldsymbol{O}$ stands for the acoustic input. The computation of $P(\boldsymbol{O}|\boldsymbol{w})$ is referred to as the acoustic model and the estimation of $P(\boldsymbol{w})$ as the language model (LM) [1, 2].

Most of the progress on LMs was made on large vocabulary dictation tasks (often referred to as Large Vocabulary Continuous Speech Recognition, LVCSR) such as the Wall Street Journal corpus [3] where read speech, which is grammatically correct, has to be recognized. In this case, an abundance of written texts is available for training and the result does not have to be analyzed syntactically and semantically. Thus it is possible to optimize the recognizer to find the word chain that matches best to the spoken word sequence, disregarding punctuation, bold faces, and paragraphs.

Things become a little bit different, when one looks at so called conversational dialogue systems, e.g. systems that have to recognize *and* understand spontaneous speech. The speech recognizer is now only one module in a larger system and its output is no longer the final result but the input to further processing stages. Figure 1 depicts the architecture of our EVAR train timetable system [4], which is a standard architecture for an information retrieval dialogue system.

Based on the user utterances word recognition is performed and the best word chain (e.g. *"I would like to go to Frankfurt"*), or alternatively a word hypotheses graph (WHG), is handed on to the linguistic processor. The linguistic processor extracts a set of semantic concepts (semantic attribute–value pairs) from the word recognizer result (e.g. `[goalcity:frankfurt]`) and forwards them to the dialogue manager. The dialogue manager checks whether all necessary parameters are available and, if so, sends a query to the application database. Depending on the dialogue history and the current dialogue strategy, the user is asked to confirm the parameter (e.g. *"You want to go to Frankfurt?"*) and/or another parameter is requested (e.g. *"At what time would you like to leave?"*); otherwise the result of the database search is verbalized. The generated message is then synthesized by a text–to–speech module and played to the user over the telephone line.

Normally, the word chain contains no additional information, such as prosodic information. However, even in the context of a comparatively simple application, such as an automatic train timetable information system, additional

[*] This work was funded by the German Federal Ministry of Education, Science, Research and Technology (*BMBF*) in the framework of the VERBMOBIL Project under Grant 01 IV 701 K5, in the framework of the SmartKom project under Grant 01 IL 905 K7 and by the DFG (German Research Foundation) under contract number 810 939-9. The responsibility for the content lies with the authors.

[†] The author is now with Sympalog Speech Technologies, Germany

[‡] The author did his work at the Lehrstuhl für Künstliche Intelligenz (Informatik 8) at the same University; he is now with Datev, Germany

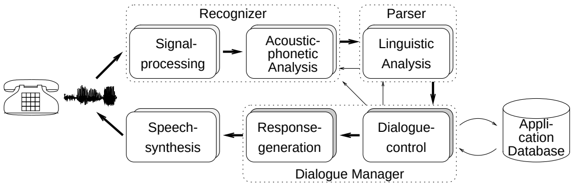

Fig. 1. The basic architecture of EVAR

information may be important. Consider, for example, the following user utterances:

U1: *"Of course not on Monday."*

U2: *"Of course not. On Monday!"*

The question whether a phrase boundary occurred after the word *"not"* is of considerable importance for the semantic interpretation of the word sequence *"of course not on Monday"*, and for determining the next system utterance. For example, either of the following two utterances may be appropriate:

S1: *"What day would you like to travel?"*

S2: *"You would like to travel on Monday?"*

Selecting the wrong response (**S1** for **U2**, or **S2** for **U1**) will most certainly annoy the caller and will probably make her/him hang up. It might be argued that the correct interpretation of the word sequence could also be determined without prosodic information, if the dialogue history is taken into account. Depending on the previous system utterance, at least one of the two above interpretations could be declared illogical. This, however, involves a considerable amount of higher–level knowledge and 'intelligent' processing, whereas prosodic information in the speech signal can directly resolve the ambiguity [5, Sec. 8.4]. Furthermore, there is no reason to ignore information that may without any doubt contribute to finding the correct semantic interpretation, even if a sufficiently intelligent dialogue module is available.

Other frequent spontaneous speech phenomena which cause severe problems to speech recognizers are *disfluencies*, such as *self–repairs*, *false starts*, and *repetitions*. These are in some cases accompanied by *silent* or *filled pauses*, such as *uh*'s and *um*'s. Disfluencies constitute a problem for the parsing of spontaneous speech: they should not be processed as such but rather disregarded: in the utterance

"yes it's ok Mon uh Sunday the fourth"

the result of syntactic analysis should rather be

"yes it's ok Sunday the fourth".

Disfluencies and particularly self–repairs like in the last example often involve *word fragments* which are by definition *out–of–vocabulary (OOV) words*, i.e. words that are not part of the predefined recognition vocabulary, because a word may be cut off mid–word, or even mid–syllable, and the recognition lexicon cannot contain all the word fragments of its words (of course — in rare cases — a fragment can happen to be a lexical entry; this can happen more often in languages like German where word compounds are quite frequent). The OOV problem severely impairs the applicability of speech recognition technology to many real–world tasks. The OOV word problem has been looked at in LVCSR, because an increase of approximately 1.5–2.5 errors per OOV word has been observed by several authors (see for instance [6, 7, 8]). Including an explicit OOV word as a lexical entry can reduce these additional errors. In an ASU system, however, it is not only important to reduce the recognition errors, but to know the semantic category of the OOV word, in order to react appropriately. For instance in the EVAR domain (and assuming that names of persons and the desired city are not in the lexicon) the user utterance

"Hello, my name is Schultz, I want to go to Rossau"

can easier be processed, if the recognizer provides as the best word chain

"Hello, my name is OOV–last–name, I want to go
to OOV–city"

rather than

"Hello, my name is OOV, I want to go to OOV".

In this paper we want to deal with these three additional information sources that can enrich the word chain, i.e. prosodic, repair, and OOV information. Adding this information to the interface between recognition and linguistic analysis can significantly improve the results of the overall system. We want to concentrate on the use of LMs to achieve these tasks. We report on four aspects of our own use of n–grams within the EVAR system and the VERBMOBIL speech–to–speech translation system [9, 10, 11]:

1. Classification of prosodic events in a WHG
2. Integrated recognition of words and phrase boundaries
3. Processing of speech repairs
4. Detection and semantic classification of OOV words

The rest of the paper is organized as follows: Section 2 outlines the various types of prosodic phenomena which have been recognized and classified with the use of LMs. The annotation scheme is described. Section 3 introduces basic concepts of category based stochastic language models and how they can be applied to classification tasks. In Section 4 classification of prosodic events and integrated recognition of words and phrase boundaries is introduced. The processing of speech repairs is described in Section 5 and the processing of OOV words in Section 6. Finally, conclusions and suggestions for future work are given.

2. PHENOMENA AND ANNOTATION

In this paper, we want to deal with the following two prosodic phenomena, for which we give examples taken from the VERBMOBIL scenario (appointment scheduling):

Boundaries:
"Fünfter geht bei mir, nicht aber neunzehnter." vs.
"Fünfter geht bei mir nicht, aber neunzehnter." i.e.
"The fifth is possible for me, but not the nineteenth." vs.
"The fifth is not possible for me, but the nineteenth would be OK."

Accentuation:
"Ich fahre doch am Montag nach Hamburg." vs.
"Ich fahre DOCH am Montag nach Hamburg." i.e.
"I will go on Monday to Hamburg." vs.
"I will go on Monday to Hamburg after all."

These are minimal pairs, which demonstrate that linguistic analysis is supported by prosodic markings (and sometimes can only be done with the help of prosody). Unfortunately there is no one–to–one mapping of prosodic classes and linguistic structure (luckily the correspondence is nevertheless very high [12]).

2.1. Boundaries

Consider the following excerpt from a real VERBMOBIL turn (translated into English), where

<A>	stands for breathing,
w<L>	for unusual lengthening of word *w*,
<P>	for a pause,
B*i*	for an acoustic–prosodic boundary
D3	for a dialogue act boundary, and
M3	for a syntactically motivated boundary:

(see below for details w.r.t. the boundary classes)
"... M3 D3 *well then I'm not present at all* B3 M3 D3 <A> *and in the*<L> B9 <P> *thirty fourth week* B3 M3 <P> <A> *that would be* B3 <P> *Tuesday* B2 *the twenty third* B3 <A> *and Thursday the twenty fifth* M3 D3 <P> *..."*

Acoustic–prosodic Boundaries
Clearly, a classifier which segments this turn based only on acoustic prosodic information, like length of a pause between words, might give the linguistic analysis boundaries which hinder rather than help (like the boundary between *"in the"* and *"thirty"*). We distinguish therefore between B0: normal word boundary; B2: intermediate phrase boundary with weak intonational marking; B3: full boundary with strong intonational marking, often with lengthening; B9: 'agrammatical' boundary, e.g., hesitation or repair. Thus we can distinguish between prosodic boundaries which correspond to the syntactic structure and others which contradict the syntactic structure. However we still have the problem that syntactic boundaries do not have to be marked prosodically. A detailed syntactic analysis would rather have syntactic boundaries irrespective of their prosodic marking, e.g., it needs to know about B9 and B0 in order to favor continuing the ongoing syntactic analysis rather than assuming that a sentence equivalent ended and a new analysis has to be started. Depending on — among other things — the speaker style, the speaker is sometimes inconsistent with his/her prosodic marking. In the example above, the intermediate boundary between *"Tuesday"* and *"the twenty third"* is clearly audible, whereas there is no audible boundary between *"Thursday"* and *"the twenty fifth"*. Syntactic phrasing is — besides by the prosodic marking — also indicated by word order. We recognize these acoustic–prosodic boundaries with classifiers (neural networks) based on acoustic–prosodic features [13, 10]. We want to recognize the different linguistic levels of boundaries with LMs which look at the word order.

Syntactic–prosodic Boundaries
For the LM training we have the demand for large training databases. The marking of perceptual labels is very time consuming, since it requires listening to the signal. We therefore developed a rough syntactic–prosodic labeling scheme, which is based purely on the orthographic transliteration of the signal, the so called M system. The scheme is described in detail in [12]. It classifies each turn of a spontaneous speech dialogue in isolation, i.e. does not take context (dialogue history) into account. Each word is classified into one of 25 classes in a rough syntactic analysis. For the purpose of the paper, it suffices to look at two different mappings into major classes:

1. M3: clause boundary (between main clauses, subordinate clauses, elliptic clauses, etc.), M0: no clause boundary;

2. S0: no boundary, S1: at particles, S2: at phrases, S3: at clauses, S4: at main clauses and at free phrases.

Dialogue Act Boundaries
Even less labeling effort and formal linguistic training is required if we label the word boundaries according to whether they mark the end of a pragmatic unit. We refer to these boundaries as dialogue act boundaries. Dialogue acts are defined based on their illocutionary force, i.e. their communicative intention, cf. [14]. Dialogue acts are e.g. 'greeting', 'confirmation', and 'suggestion'; a definition of dialogue acts in VERBMOBIL is given in [15], [16]. In parallel to the B and M labels, we distinguish between D3: dialogue act boundary, and D0: no dialogue act boundary.

2.2. Phrase Accents

We distinguish between four different types of syllable based phrase accent labels which can easily be mapped onto word based labels denoting if a word is accented or not:

PA: primary accent; SA: secondary accent; EC: emphatic or contrastive accent; A0: any other syllable (not labeled explicitly). Since the number of PA, SA, EC labels is not large enough to distinguish between them automatically, we only ran experiments trying to classify 'accented word' (A3 = {PA, SA, EC}) vs. 'not accented word' (A0). In the VERBMOBIL domain, the number of emphatic or contrastive accents is not very large. In information retrieval dialogues this could easily change, if there is a large number of misunderstandings and corrections. Again, these are the basis for our 'acoustic model'. For the LM, we developed a rule–based system which — starting with the M boundaries — predicts for each word between two boundaries, whether it carries the phrase accent, based on the part–of–speech (POS) sequence in the syntactic phrase. The system is described in [17].

3. N–GRAM LANGUAGE MODELS

In this section, the problem of estimating stochastic language models $P(\boldsymbol{w})$ for sentences $\boldsymbol{w} = w_1 \dots w_m$ of words w_i from a finite vocabulary $\mathcal{W}$ is addressed. The joint distribution $P(\boldsymbol{w})$ can be decomposed by the chain rule

$$\begin{aligned} P(\boldsymbol{w}) &= P(w_1) \prod_{i=2}^{m} P(w_i|\boldsymbol{w}_1^{i-1}) \\ &= P(w_1) \prod_{i=2}^{m} P(w_i \mid w_1 \dots w_{i-1}) \end{aligned}$$

into a product of conditional word probabilities. An *n–gram language model* is obtained if only sub–sequences of length n (*n–grams*) are taken into account, that is, the history is restricted to $n-1$ words:

$$P(\boldsymbol{w}) \approx P(w_1) \prod_{i=2}^{m} P(w_i|\boldsymbol{w}_{i-n+1}^{i-1})$$

The straightforward approach is to replace the conditional n–gram probabilities by their maximum likelihood estimates

$$\hat{P}(w_i|\boldsymbol{w}_{i-n+1}^{i-1}) = \frac{\#(\boldsymbol{w}_{i-n+1}^{i})}{\#(\boldsymbol{w}_{i-n+1}^{i-1})}$$

where the function $\#(\cdot)$ gives the frequency of occurrences of its argument in the training text. Typical values of n in speech recognition applications are $n = 2$ (*bigram*) and $n = 3$ (*trigram*).

Unfortunately, the frequency ratios are far from being reliable probability estimates, even in the case of small values for n. In particular, $\hat{P}(w_i|\boldsymbol{w}_{i-n+1}^{i-1})$ degenerates to zero if the n–gram $\boldsymbol{w}_{i-n+1}^{i}$ was never observed in the training data. An even larger problem arises as soon as the denominator $\#(\boldsymbol{w}_{i-n+1}^{i-1})$ of the ML estimate expression turns to zero. As a consequence, the raw ML estimates have to be smoothed; non–zero probabilities have to be assigned to *unseen* word sequences, and that *probability mass* has to be taken from non–zero ML estimates. There are two basic strategies that are employed for this purpose: *Backing–off* approaches [18] and *interpolation* strategy schemes [19].

3.1. Category based n–Grams

Where for LVCSR huge amounts of written text are often available, the training data for ASU systems have to be transcribed from recorded dialogues. This is a very expensive task. For instance, the EVAR training set and cross–validation set together contain only about 60,000 words (2,300 different). The number of parameters in n–gram models can be drastically reduced, if *word categories* (or *word classes*) are introduced. These can be based on syntactic, semantic, and pragmatic knowledge, or they can be determined automatically with the use of clustering algorithms. Here, only categories $\mathcal{Z} = \{\mathcal{Z}_1, \mathcal{Z}_2, \dots, \mathcal{Z}_D\}$ are considered that do not overlap and build a partition of the vocabulary $\mathcal{W}$, that is, each word sequence $\boldsymbol{w} = w_1 \dots w_m$ corresponds to a unique sequence of word categories $\boldsymbol{z} = z_1 \dots z_m, z_i \in \mathcal{Z}$. The probability of observing a word sequence $\boldsymbol{w}$ can then be denoted as

$$P(\boldsymbol{w}) \approx P(z_1)P(w_1 \mid z_1) \prod_{i=2}^{m} P(z_i \mid \boldsymbol{z}_{i-n+1}^{i-1})P(w_i \mid z_i)$$

Any type of n–gram can be used to model the probabilities of category sequences. Additionally, the conditional probability for a symbol given a category has to be estimated. This is usually done according to the relative frequency of the words belonging to each category, or with the same smoothing techniques as for word based models.

The use of word categories can significantly improve the robustness of language model training. Manually constructed word categories, however, have to be carefully selected (consider, for example, a single category 'number' for the train timetable information domain; because of different ranges for hours and minutes, this might be a bad choice [20]).

We build a category system that contains the following word categories:

- all relevant predefined word categories, i.e. word categories which contain words that are sufficiently frequent in the training data. As indicated above, this is very application dependent. Example categories are 'first name', 'last name', 'city name', 'region', 'day of week', and 'month';

- a category of its own for each sufficiently frequent word which is not included in one of the manually designed word categories, and

- a single word category for all remaining words which are not included in one of the manually designed word categories.

3.2. Classification with Language Models

Let w_i again be a word out of a vocabulary where i denotes the position in the utterance (the approach works as well, if w_i denotes a category). v_i denotes a symbol out of a predefined set $\mathcal{V}$ of prosodic symbols. These can be for example {M3, M0}, {A3, A0}, or a combination of both {M0A0, M0A3, M3A0, M3A3} depending on the specific classification task. For example, $v_i = \text{M3}$ means that the i^{th} word in an utterance is succeeded by a clause boundary.

Classification is done with the Bayes' Rule by computing the posterior probability for the occurrence of a prosodic symbol $V_i \in \mathcal{V}$, given a string where words and prosodic labels alternate:

$$P(v_i = V_i | w_1 v_1 \ldots w_{i-1} v_{i-1} w_i w_{i+1} v_{i+1} \ldots w_m v_m) \\ = \frac{P(w_1 v_1 \ldots w_{i-1} v_{i-1} w_i V_i w_{i+1} v_{i+1} \ldots w_m v_m)}{\sum_{V_i \in \mathcal{V}} P(w_1 v_1 \ldots w_{i-1} v_{i-1} w_i V_i w_{i+1} v_{i+1} \ldots w_m v_m)}$$

According to the last equation we need to model the following a priori probability:

$$P(w_1 v_1 w_2 v_2 \ldots w_m v_m)$$

When determining the appropriate label V_i to substitute v_i, the labels at positions v_{i-k} and v_{i+k} are not known ($k = 1, 2, \ldots$). To simplify the computation, we approximate

$$P(w_1 v_1 w_2 v_2 \ldots w_m v_m) \approx \\ P(w_{i-n+2} \ldots w_{i-2} w_{i-1} w_i v_i w_{i+1} w_{i+2} \ldots w_{i+n-2})$$

and represent the distribution by n–grams which are estimated on strings of words and prosodic symbols.

If one wants to classify v_i in a WHG instead of a word chain, the exact solution would be a weighted sum of all probabilities P_{v_i} computed on the basis of all the possible contexts, i.e. all possible paths through w_i. However, this does not seem to be feasible under real–time constraints. Instead we classify v_i based on the locally best path through w_i by looking at $n-2$ predecessors and successors of w_i.

4. CLASSIFICATION OF ACCENTS AND BOUNDARIES WITH LMS

We have defined the classes (Section 2) and a classifier (Section 3.2). The classifier looks at the word sequence and not at the acoustic evidence. Classification with neural networks based on acoustic evidence is described in [10, 13]. As different *'understanding modules'* in VERBMOBIL use our classification results and look at different resolutions (S vs. M vs. D) [9] and since classification errors have different effects depending on whether a prosodic event is not found (miss) or its complement is wrongly classified as a prosodic event (false alarm), we pass on acoustic based and word sequence based classification separately (note that in a previous version of our prosody module we combined the acoustic and word sequence based classification [13]). In Tables 1 and 2 we present the recall, i.e., $correct/(correct + miss)$, for defined classes. Precision, i.e., $correct/(correct + false\ alarm)$ can be computed from the numbers provided. The results are achieved on the basis of the spoken word chain, i.e. simulating a perfect word recognizer.

	set	M3	M0	A3	A0	D3	D0
G	# train	27k	126k	103k	174k	15k	99k
	# test	5k	24k	3k	5k	5k	26k
	recall	**86**	**97**	**87**	**92**	**80**	**96**
E	# train	16k	53k	–	–	–	–
	# test	2k	6k	–	–	–	–
	recall	**83**	**94**	–	–	–	–
J	# train	–	–	–	–	14k	94k
	# test	–	–	–	–	1k	8k
	recall	–	–	–	–	**92**	**99**

Table 1. LM classification: Recall in percent for syntactic–prosodic boundaries M, rule–based accents A, and dialogue act boundaries D in the three languages of the VERBMOBIL system: German (G), English (E) and Japanese (J); number of cases is given for train and test

reference		**recognized**				
German						
label	#	S0	S1	S2	S3	S4
S0	24286	**89**	2	5	2	2
S1	1408	8	**81**	4	2	5
S2	1014	15	3	**69**	3	10
S3	622	8	2	5	**73**	12
S4	3640	4	5	6	6	**79**
English						
label	#	S0	S1	S2	S3	S4
S0	5771	**89**	1	6	2	2
S1	169	7	**64**	17	0	12
S2	900	5	3	**83**	2	8
S3	145	7	1	7	**71**	14
S4	1066	3	8	9	3	**76**

Table 2. Recall in percent for the five S classes

Integrated Recognition of Words and Boundaries

The approach just presented has the disadvantage that knowledge about the position of phrase boundaries cannot be used for determining the spoken word sequence. As has been pointed out by other authors [21, 22], information about the syntactic structure of an utterance can improve the word recognition result. Taking a look at our VERBMOBIL test database with respect to the occurrence of unseen word pairs, we found that of all pairs of neighboring words which are *within* phrases that are delimited by M3 phrase boundaries, only 14% have never been observed in the training sample. The same ratio for word pairs *across* phrase boundaries is 38%. Any standard n–gram language model will provide lower probabilities for word transitions that have not been observed in the training data. That is, language model probabilities across phrase boundaries are *systematically underestimated* by word–based language models.

In our integrated approach for word–and–boundary recognition, utterances are not modeled as unstructured sequences of words, as in traditional word recognizers, but as sequences of words and boundaries. Both words and boundaries are therefore integrated in a single language model. Furthermore, suitable HMMs have to be provided for phrase boundaries. We developed the LMs on the basis of the M0/M3 classes. It is worth noting that 59% of the M3 boundaries are marked by a pause or a non–verbal sound, and that 67% of all pauses and non–verbals coincide with an M3 boundary ([20]). We thus have to provide HMMs to model M3 boundaries with and without pauses and non–verbals and HMMs to model pauses and non–verbals without M3 boundary. For phrase boundaries that do not coincide with a filled pause or a non–verbal, we use a one–state HMM that always consumes a single time frame. Table 3 shows the complete inventory of boundary HMMs together with their non–boundary equivalents.

M3 boundary model	non–boundary equivalent	# HMM states
[M3]	(none)	1
[-M3-]	[-]	3
[—M3—]	[—]	9
[M3:um]	[um]	9
[M3:NV]	[NV]	9
[M3:breathing]	[breathing]	9

Table 3. The inventory of boundary HMMs and their non–boundary equivalents. The HMM training for all these models is performed in a partially unsupervised manner as described in [20]

The integrated language model for words and phrase boundaries is a regular n–gram model which is constructed as follows:

- M3 boundary models are treated like words
- all M3 boundaries are included in a single, additional word category M3
- non–boundary models for pauses and non–verbal phenomena are treated as random events that do not depend on the surrounding word context. They are ignored, when the probability of the following word is calculated.

In Figure 2, the integrated word–and–boundary language model is illustrated with an example utterance.

Table 4 shows the results for the baseline and the integrated word–and–boundary recognizer. There is a small improvement in word error rate and part of the syntactic structure of the utterance is recognized ‘for free’, i.e. the interface to the understanding module contains more information with no computational overhead (in fact the computation is slightly faster). The approach is described in detail in [20, 23]. There a hybrid recognizer that also uses acoustic boundary evidence is described as well.

The results are not comparable to the ones presented in Table 1, because those were achieved on the basis of the spoken word chain. When using the output of the baseline word recognizer instead of the spoken word chain, precision and recall for the M3 and M0 classes are practically identical for the sequential and the integrated approach.

system	WER	recall	prec.	RTF
baseline word rec.	23.8%	—	—	4.1
integrated word–and–boundary rec.	22.9%	74.5%	75.7%	4.0

Table 4. Word error rates (WER), recall and precision rates for M3 phrase boundaries, and real time factor (RTF) on the VERBMOBIL test sample

5. REPAIRS

In the German part of the VERBMOBIL corpus, 21% of all turns contain at least one repair. Most of them (82%) are so called modification repairs and we therefore concentrate on this type of repairs (for a detailed analysis of the different kinds of repairs see [24]). Modification repairs correct part of the whole sentence, but do not change the syntactic construction. We define repetitions as a special case of modification repairs, where the corrected part and the correction are identical. Commonly each repair is segmented in the four parts *reparandum* (RD), *editing term* (ET), *interruption point* (IP), and *reparans* (RS); an example is given in Figure 3:

- RD: the ‘wrong’ part of the utterance
- IP: boundary marker at the end of the RD

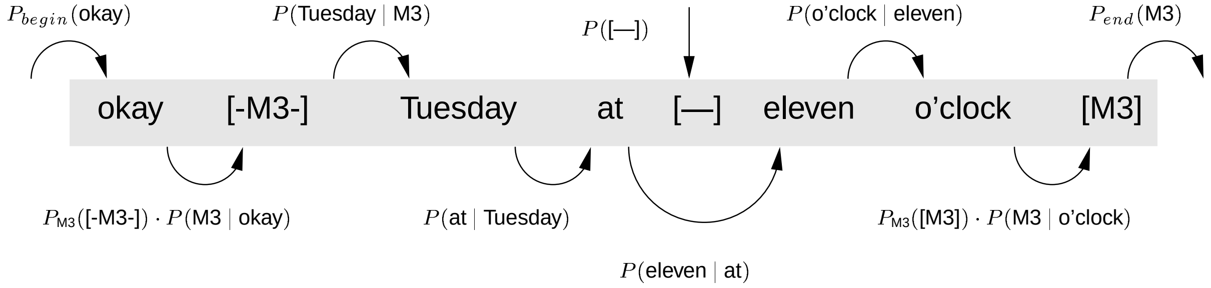

Fig. 2. The integrated word–and–boundary language model (in the case of a bigram–based recognizer) illustrated with the example utterance *"okay — Tuesday at — eleven o'clock"* (the dashes indicate silent pauses). The correct sequence of word and boundary models and the corresponding bigram probabilities are given in the figure. All M3 boundary models (e.g. [-M3-] for a boundary which is marked by a silent pause and [M3], which consumes only one time frame) are in a single language model category M3; the category–dependent emission probabilities for M3 models are denoted $P_{\text{M3}}(\cdot)$.

- ET: special phrases, which indicate a repair like *"well"*, *"I mean"* or filled pauses such as *"uhm"*, *"uh"* (optional, most of the time missing)
- RS: the correction of the RD

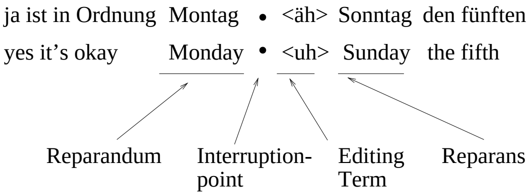
ja ist in Ordnung Montag • <äh> Sonntag den fünften
yes it's okay Monday • <uh> Sunday the fifth
Reparandum
Interruption-point
Editing Term
Reparans

Fig. 3. A repair example

Modification repairs have a strong correspondence between RD and RS. We can measure this in terms of length of RD and RS and POS replacements. For almost all POS categories, the speakers prefer to modify a word in the RD with a word which belongs to the same POS category in the RS. Thus there is no need for a complete syntactic analysis to detect and correct most modification repairs even if repairs are characterized by violation of syntactic and semantic well–formedness [25]. We implemented a statistical approach as a filter process between the speech recognition engine and the syntactic parser. Starting with the WHG produced by the word recognizer, a prosodic module detects possible IPs. For each of these IPs, a stochastic model tries to find an appropriate repair by guessing the most probable segmentation.

Detection of Interruption Points

The prosodic module classifies each word boundary in the WHG as a regular or an irregular boundary. Irregular boundaries are seen as hypotheses for IPs (details are given in [10]). Note that IPs are a mixture of B2/B3 and B9 boundaries (Section 2.1), since they can either coincide with syntactic boundaries or they can be 'agrammatical'. On the other hand, hesitations can be B9 boundaries and yet do not necessarily mark repairs. A classification of a subsample of the VERBMOBIL database with neural networks and 559 IPs vs. 51,486 'normal' word boundaries (i.e., a relation of 1:100!) yielded the results shown in Table 5.

Segmentation

Repair processing is seen as a statistical machine translation (SMT) problem [26] where the RD is a translation of the RS. The SMT approach assumes that a speaker who produces the source sentence S originally wants to produce the target sentence T. Transferring this approach to repair processing, the source sentence is represented by the RD and the target sentence is equivalent to the RS. SMT defines a scoring function for a pair (S, T) which can be adopted for repair processing without further changes:

$$P(RD|RS) = \sum_a P(RD, a|RS)$$

a is the alignment, which describes the link between words in RD and RS. The probabilities are estimated with a linear interpolation of n–grams for the words, the corresponding POS tags, and the semantic classes. Details are given in [24, 27].

Integration into the VERBMOBIL System

The repair module is integrated in the VERBMOBIL system on top of the prosodically annotated WHG from the recognizer. For each path through the WHG that contains an IP hypothesis, all possible segmentations, i.e., all possible (RD, RS) pairs, must be scored. In practice we reduce this set to pairs, where RD and RS are at most four words long, because we found that this restriction holds for 96% of all repairs in the VERBMOBIL corpus. ETs are characterized by a closed list of short phrases. Thus if after an IP such a phrase is found, it is skipped to build the (RD, RS) pair. If the score of a pair is above a heuristic threshold, the pair is accepted as a repair and an alternative path is inserted in the WHG. The resulting WHG is finally analyzed by a stochastic parser, which selects according to its model the best scored path and therefore can accept or reject the repair.

	detection		correct RS	
	recall	prec.	recall	prec.
prosodic classifier	90%	3%	—	—
repairs without word fragments	48%	77%	47%	76%
repairs including word fragments	70%	86%	61%	84%

Table 5. Results for repair processing

Discussion of the Results

Table 5 shows the results of the repair process with the assumption that we have a perfect recognizer that produces no word errors and marks every word fragment. The 'detection' column shows the results for the repair identification task. The 'correct RS' column presents the same numbers for the correct segmentation. A segmentation is defined as 'correct' if RD and ET are identified. In some cases within complex repairs (repairs within repairs), RD and ET are not identified correctly but, if these segments are removed from the input, the resulting string is the intended word sequence.

One major problem in handling self repairs are word fragments. They occur often at the end of the RD and constitute an important repair signal. But current state of the art speech recognizer cannot detect word fragments. So any analysis based on word fragment information does not reflect the performance in a real speech system. Thus, we perform two tests: One with word fragment information and one where we exclude turns with fragments.

The first row in this table shows the results for prosodic IP detection. One can see the problem of a solely prosody based repair detection. The neural network recognizes 90 percent of all repairs, but produces a lot of false alarms as indicated by the bad precision. The reason is not a worse classifier but a principle problem. At first the event IP is very rare in contrast to the event 'no interruption', which is a bad precondition for a two class classifier. And secondly, the prosodic features that are used to mark the IP can be observed in many situations that constitute no repair.

But as can be seen in the following rows the repair search process can eliminate many of those false alarms. When we count only turns without fragments, we detect and correct almost half of the repairs. The last row shows the strong impact of fragments to repair processing. By using fragment information recall and precision of detection and correction increase.

We believe that modeling modification repairs as a translation process is an promising approach to repair processing. The formal description in terms of statistical machine translation opens a great variety for further model improvements. The main unsolved problems are word fragments and fresh starts.

6. DETECTION AND CLASSIFICATION OF OOV WORDS

In [28] we presented an approach for the detection of OOV words which implicitly provides information on the word category. This involves the integration of both detection *and* classification of OOV words directly into the recognition process of an HMM–based word recognizer. With our approach, acoustic information as well as language model information can be used for the purpose of classifying OOV words into different word categories. Currently the same acoustic models are used for all OOV words; only language model information contributes to the assignment of a category to each.

The basic idea behind our approach is to build language models for the recognition of OOV words that are based on a system of word categories. Emission probabilities of OOV words are then estimated for each word category. Even if we include in our vocabulary all words of a category that were observed in the training sample, there is still a certain probability of observing other new words of the same category in an independent test sample or in future utterances. This probability can be estimated from the training sample itself. Details on the calculation of the OOV emission probabilities were given in [28]; an improved version of the algorithm can be found in [20]. Figure 4 shows the principle of this estimation technique for the category *city* of the EVAR sample.

For most of our linguistically motivated word categories, the OOV probability is 0, because they describe a finite set of words. In the time table inquiry domain there are 5 word categories that are practically infinite (e.g. *city, region, last name*). In addition, a category for rare words has been defined that do not fall under any other category (OOV probability 73%) and another for garbage (e.g. word fragments, OOV probability 100%).

After integrating OOV probabilities into the language model, the latter has to be combined with one or several acoustic models for OOV words. Simple 'flat' acoustic models can be used for this purpose as well as more enhanced models based on phone– or syllable–grammars.

The results for the VERBMOBIL domain are summarized in Table 6. The total number of OOV words in the test sample was 132, i.e. an OOV rate of 2.8%. At the first glance, the overall recall and precision rates for OOV words of 28% and 32% are rather disappointing. Interestingly,

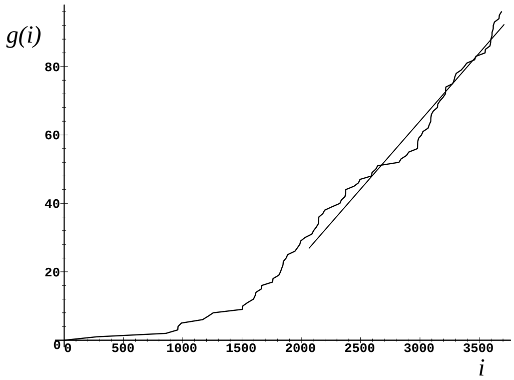

Fig. 4. Estimation of the current OOV word probability for word category *city*. The function g gives the number of words in category *city* up to the ith word of the training sample that would have been OOV if we had redefined the vocabulary after each observed word. The slope of the linear approximation is an estimation of the OOV probability of category *city*

however, the word error rate after including OOV words drops from 22.5% to 22.1%. This is due to the fact that OOV false alarms occur mainly at those parts of utterances where word recognition errors would also have occurred without OOV models in the vocabulary. These results are comparable to those achieved in [8].

7. CONCLUSION AND OUTLOOK

In the field of speech recognition, stochastic n-grams are widely used for the estimation of the probability of word strings. Their success is mainly due to an unique combination of favorable features: n-grams can be estimated easily from transcribed speech or text data and their structure makes it possible to integrate them into a time-synchronous recognition process. In this paper, we showed how these characteristics facilitate the processing of complex speech phenomena, like prosodic events and speech repairs as well as the application of n-grams to the recognition and classification of OOV words. We demonstrated that the structure of n-grams supports the integration of the detection of prosodic boundaries into the speech recognizer and showed that this approach can also improve the speed and accuracy of word recognition.

We expect that further improvements may be gained by a better integration of the different modules. For instance, the OOV word detection and classification has not been evaluated yet in conjunction with the repair processing module. Currently, all word fragments fall into the garbage category of the OOV module. The repair processing module may receive more specific information if word fragments would be classified into different categories, e.g. if a separate category for word fragments of weekdays (*"yes it's ok Mon"*) would exist. Another important research area which has not been mentioned in this paper so far is the classification of different *emotions*, e.g. *anger*, and *user states*, e.g. *helplessness*. Emotions and user states are expressed by prosodic, lexical, syntactic/semantic, and illocutionary means [29]. At least some of these means can be modeled with n–grams. In the near future, we plan to evaluate, if the integration of the information sources provided by all modules described in this paper can improve the current performance on this task.

system	baseline	OOV–extended
WER	22.5%	22.1%
RTF	3.8	3.9
recall total	0%	28%
precision total	–	32%
recall LAST_NAMES	0%	35%
precision LAST_NAMES	–	68%

Table 6. Evaluation of the OOV–extended recognizer for the VERBMOBIL domain. The recall and precision rates are given for all OOV words (recall total and precision total) and for OOV words from word category LAST_NAMES (recall LAST_NAMES and precision LAST_NAMES)

8. REFERENCES

[1] F. Jelinek, "Continuous Speech Recognition by Statistical Methods," *Proceedings of the IEEE*, vol. 64, no. 4, pp. 532–556, 1976.

[2] Frederick Jelinek, *Statistical Methods for Speech Recognition*, MIT Press, Cambridge, 1997.

[3] D. Paul and J. Baker, "The Design for the Wall Street Journal–based CSR Corpus," in *Proc. Speech and Natural Language Workshop*, San Mateo, 1992, pp. 1–5, Morgan Kaufmann Publishers Inc.

[4] F. Gallwitz, M. Aretoulaki, M. Boros, J. Haas, S. Harbeck, R. Huber, H. Niemann, and E. Nöth, "The Erlangen Spoken Dialogue System EVAR: A State–of–the–art Information Retrieval System," in *Proc. of the 1998 Int. Symposium on Spoken Dialogue (ISSD 98)*, Sydney, 1998, pp. 19–26.

[5] R. Kompe, *Prosody in Speech Understanding Systems*, Lecture Notes for Artificial Intelligence. Springer–Verlag, Berlin, 1997.

[6] I. L. Hetherington, *A Characterization of the Problem of New, Out–of–Vocabulary Words in Continuous–Speech Recognition and Understanding*, Ph.D. thesis, MIT, Cambridge, USA, 1995.

[7] D. Pallet, J. Fiscus, and M. Fisher, "1994 Benchmark Tests for the ARPA Spoken Language Program," in *Proc. ARPA Spoken Language Systems Technology Workshop*, San Mateo, 1995, Morgan Kaufmann Publishers Inc.

[8] P. Fetter, *Detection and Transcription of Out–of–Vocabulary Words in Continuous–Speech Recognition*, Ph.D. thesis, Technische Universtität Berlin, Germany, 1998.

[9] W. Wahlster, Ed., *Verbmobil: Foundations of Speech-to-Speech Translations*, Springer, Berlin, 2000.

[10] A. Batliner, A. Buckow, H. Niemann, E. Nöth, and V. Warnke, "The Prosody Module," In Wahlster [9], pp. 106–121.

[11] J. Spilker, M. Klarner, and G. Görz, "Processing Self–Corrections in a Speech–to–Speech System," In Wahlster [9], pp. 131–140.

[12] A. Batliner, R. Kompe, A. Kießling, M. Mast, H. Niemann, and E. Nöth, "M = Syntax + Prosody: A syntactic–prosodic labelling scheme for large spontaneous speech databases," *Speech Communication*, vol. 25, no. 4, pp. 193–222, 1998.

[13] E. Nöth, A. Batliner, A. Kießling, R. Kompe, and H. Niemann, "Verbmobil: The Use of Prosody in the Linguistic Components of a Speech Understanding System," *IEEE Trans. on Speech and Audio Processing*, vol. 8, pp. 519–532, 2000.

[14] J.R. Searle, *Speech Acts. An Essay in the Philosophy of Language*, Cambridge University Press, Cambridge, 1969.

[15] S. Jekat, A. Klein, E. Maier, I. Maleck, M. Mast, and J. Quantz, "Dialogue Acts in Verbmobil," Verbmobil Report 65, 1995.

[16] M. Mast, E. Maier, and B. Schmitz, "Criteria for the Segmentation of Spoken Input into Individual Utterances," Verbmobil Report 97, 1995.

[17] A. Batliner, M. Nutt, V. Warnke, E. Nöth, J. Buckow, R. Huber, and H. Niemann, "Automatic Annotation and Classification of Phrase Accents in Spontaneous Speech," in *Proc. European Conf. on Speech Communication and Technology*, Budapest, 1999, vol. 1, pp. 519–522.

[18] S. Katz, "Estimation of Probability from Sparse Data for the Language Model Component at a Speech Recognizer," *IEEE Trans. on Acoustics, Speech and Signal Processing*, vol. ASSP-35, no. 3, pp. 400–401, 1987.

[19] F. Jelinek, "Self-Organized Language Modeling for Speech Recognition," in *Readings in Speech Recognition*, A. Waibel and K. F. Lee, Eds., pp. 450–506. Morgan Kaufmann Publishers Inc., San Mateo, 1990.

[20] F. Gallwitz, *Integrated Stochastic Models for Spontaneous Speech Recognition*, Studien zur Mustererkennung. Logos Verlag, Berlin, (to appear).

[21] M. Ostendorf, C.W. Wightman, and N.M. Veilleux, "Parse Scoring with Prosodic Information: an Analysis/Synthesis approach," *Computer Speech & Language*, vol. 7, no. 3, pp. 193–210, 1993.

[22] F. Jelinek and C. Chelba, "Putting Language Into Language Modeling," in *Proc. European Conf. on Speech Communication and Technology*, Budapest, 1999, vol. 1, pp. KN–1–KN–5.

[23] F. Gallwitz, H. Niemann, E. Nöth, and V. Warnke, "Integrated Recognition of Words and Prosodic Phrase Boundaries," *Speech Communication*, vol. 36, no. 1-2, 2002.

[24] J. Spilker, A. Batliner, and E. Nöth, "How to Repair Speech Repairs in an End-to-End System," in *Proc. ISCA Workshop on Disflueny in Spontaneous Speech*, R. Lickley and L. Shriberg, Eds., Edinburgh, 2001, pp. 73–76.

[25] W. Levelt, "Monitoring and self–repair in speech," *Cognition*, vol. 14, pp. 41–104, 1983.

[26] P.F. Brown, J. Cocke, S.A. Della Pietra, V.J. DellaPietra, F. Jelinek, J.D. Lafferty, R.L. Mercer, and P.S. Roossin, "A Statistical Approach to Machine Translation," *Computational Linguistics*, vol. 16, no. 2, pp. 79–85, 1990.

[27] J. Spilker, *Behandlung spontansprachlicher Reparaturen in einem Sprachverarbeitungssystem*, Ph.D. thesis, Universität Erlangen-Nürnberg, Germany, 2001.

[28] F. Gallwitz, E. Nöth, and H. Niemann, "A Category Based Approach for Recognition of Out-of-Vocabulary Words," in *Proc. Int. Conf. on Spoken Language Processing*, Philadelphia, 1996, vol. 1, pp. 228–231.

[29] A. Batliner, K. Fischer, R. Huber, J. Spilker, and E. Nöth, "How to Find Trouble in Communication," *Speech Communication*, (to appear).