Int. J. Appl. Math. Comput. Sci., 2013, Vol. 23, No. 3, 649–668
DOI: 10.2478/amcs-2013-0049

amcs

PIPELINED LANGUAGE MODEL CONSTRUCTION FOR POLISH SPEECH RECOGNITION

JERZY SAS *, ANDRZEJ ŻOŁNIEREK **

* Institute of Informatics
Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: jerzy.sas@pwr.wroc.pl

** Department of Systems and Computer Networks
Wrocław University of Technology, Wybrzeże Wyspiańskiego 27, 50-370 Wrocław, Poland
e-mail: andrzej.zolnierek@pwr.wroc.pl

The aim of works described in this article is to elaborate and experimentally evaluate a consistent method of Language Model (LM) construction for the sake of Polish speech recognition. In the proposed method we tried to take into account the features and specific problems experienced in practical applications of speech recognition in the Polish language, reach inflection, a loose word order and the tendency for short word deletion. The LM is created in five stages. Each successive stage takes the model prepared at the previous stage and modifies or extends it so as to improve its properties. At the first stage, typical methods of LM smoothing are used to create the initial model. Four most frequently used methods of LM construction are here. At the second stage the model is extended in order to take into account words indirectly co-occurring in the corpus. At the next stage, LM modifications are aimed at reduction of short word deletion errors, which occur frequently in Polish speech recognition. The fourth stage extends the model by insertion of words that were not observed in the corpus. Finally the model is modified so as to assure highly accurate recognition of very important utterances. The performance of the methods applied is tested in four language domains.

Keywords: automatic speech recognition, hidden Markov model, adaptive language model.

1. Introduction

Automatic Speech Recognition (ASR) over the past twenty years has been the challenge for researchers all over the world. Although various ASR paradigms have been proposed and investigated, the most popular approach to ASR is the one where the *Hidden Markov Model* (HMM) of speech is created using the *Language Model* (LM) and the *Acoustic Model* (AM). The LM represents the stochastic properties of the language. In the course of ASR recognition, it is used to estimate the probabilities of word sequences that can constitute fragments of utterances being recognized. The LM is typically created using a representative set of texts from the domain of ASR application. The set of texts used to construct LM is usually called the *corpus*. Different LM construction methods have been investigated in order to achieve the accepted level of user satisfaction with a spoken man–machine dialog, which can be measured empirically as the *Word Error Rate* (WER). A WER less than 5–10% must be reached in order for ASR-based software to be widely accepted (Devine *et al.*, 2007). A typical approach lies in applying a stochastic n-gram LM, but because of the insufficient amount of the available data different discounting techniques composed with *back-off* methods are proposed to obtain a smoothed LM, which assigns non-zero probabilities also to n-grams not occurring in the text corpus. In this typical approach (Goodman, 2001; Jurafsky and Matrin, 2009; Chen and Goodman, 1999; Gale and Sampson, 1995; Katz, 1987) the probability mass obtained by discounting n-gram probabilities is distributed in different ways among all non-observed n-grams taking into account $(n-1)$-gram probabilities.

Such static strategies strongly depend on the size of the vocabulary used, and in order to improve the quality of the ASR system, different adaptive (dynamic) technique are used. A class-based LM was applied by Brown *et al.* (1992), Ward and Issar (1996) or Niesler

et al. (1998). The method described by Chen and Chan (2003) or Sarukkai and Ballard (1996) introduces a trigger pair model to investigate a long distance dependent relationship. The method presented by Mikolov *et al.* (2011) is based on a combination (in the form of linear interpolation) of advanced language modeling techniques such as the class-based model, the cache model, the maximum entropy model, structured LM and others. The results of Iyer and Ostendorf (1999) suggest modelling long distance dependence using topic mixtures model.

While the accuracy of advanced commercial and experimental ASR systems for English reaches 97–98%, the accuracy achieved typically for other languages is significantly lower. The main reason of difficulties in achieving a low WER is rich inflection of the language and loose word order permitted by the language syntax. Slavonic languages are particularly difficult for ASR due to these reasons (Ziółko *et al.*, 2010; Mauces *et al.*, 2003). Because of rich inflection, the language dictionary contains much more word forms than in the case of inflectionally simple languages. Experiments described by Whittaker and Woodland (2003) show that in order to obtain a similar level of corpus coverage, the dictionary for Russian has to contain almost 7 times more words than is needed for English. Firstly, a big number of words in the dictionary leads to computational problems in the recognition process. Additionally, phonetic differences of word form pronunciations are often insignificant, which leads to problems in distinguishing word forms based on acoustic evidence. Therefore, for languages having specific properties different methods taking into account their individual features are considered.

For the Lithuanian language, Vaiciunas *et al.* (2004) proposed word clustering first and then linear interpolation of a classical model with a class-based model. Another source of difficulties in ASR for languages like Czech or Slovak or Polish is their loose word order. In these languages the word order is not so strictly imposed by the language syntax as, e.g., in English or German. If a sequence of words constitutes a syntactically and semantically correct utterance, then it is very likely that the permutation of these words also constitutes a syntactically correct phrase. As a result, the language model perplexity of Polish is much higher than that of English (Ziółko *et al.*, 2010; Jurafsky and Matrin, 2009). A typical LM based on counting n-grams appearing in the language corpus and estimating the probability of the next n-th word conditioned on the preceding sequence of $n-1$ words is less effective in supporting ASR. This is because the actual conditional probability $p(w_i|w_{i-n+1},\ldots,w_{i-1})$ is more uniformly distributed among words w_i. For the Czech and Slovak languages, Brychcin and Konopik (2011) proposed to use morphological knowledge in a class-based n-gram LM with linear interpolation.

In loose word order languages, instead of merely relying only on the sequences of words actually observed on adjacent positions in the training corpus, it seems reasonable to increase the n-gram probabilities of all word pairs that co-occur in the utterances in the language corpus. In this way, a longer context of words can be taken into account in the language model. Incorporation of a distant context into the LM has been considered in a number of publications. One of the possibilities is to take higher order n-gram models.

Experiments described by Goodman (2001) show that increasing the n-gram order up to 6 improves the perplexity. This however, requires a very huge language corpus in order for higher order n-gram probabilities to be estimated reliably. Another approach utilizes the observation that words once observed in the text are likely to be repeated again. This leads to the above-mentioned concept of a dynamic language model where the probabilities of words just observed in the text are temporarily boosted (Jelinek *et al.*, 2001).

ASR accuracy can be also improved by applying a multistage approach, where the earlier stage provides a set of alternative word sequences and the subsequent stages re-evaluate the candidate sequence scoring by applying longer distance word co-occurrence properties. This concept was applied in the Julius ASR system (Lee *et al.*, 2001) and proved to be effective also for Polish. The method presented by Piasecki and Broda (2007) exploits the concept of semantic similarities of words. It was originally proposed for handwriting recognition but can be easily adapted to ASR needs. The likelihood is boosted for sequences containing word combinations that are semantically similar each to other. The semantic similarity can be defined in various ways, but one of possibilities is to base it on word co-occurrence frequency in the language corpus. Another idea described by Kolorenc *et al.* (2006) explores the influence of multi-words (compound words) in the continuous speech recognition system of the Czech language. Multi-words are made of short words (at most three characters long) and frequently the following or the preceding word and are added to the vocabulary. Quite a different approach to ASR in Polish is presented in the work of Ziółko *et al.* (2010), where instead of the HMM the method using the Levenshtein distance is proposed. Another paper concerning ASR for the Polish language (Ziółko *et al.*, 2011) considers specific acoustic features of the language.

The adaptive approach presented in our previous paper (Sas and Żołnierek, 2011) lies in modification of typical n-gram LM. The modifications are arranged so as to boost the probabilities of n-grams consisting of words that co-occur in utterances but are not direct neighbors. The modification can be applied to any backoff LM that is based on discounting. In a typical approach (Goodman,

2001) the probability mass obtained by discounting estimated n-gram probabilities is distributed among all not observed n-grams proportionally to $(n-1)$-gram probabilities. In our approach proposed previously (Sas and Żołnierek, 2011), more probability is allocated to co-occurring n-grams at the expense of lowering the probability allocated to n-grams whose components did not occur in the same utterance in the language corpus. The factor by which the co-occurring n-gram probabilities are boosted is set so as to minimize the cross-perplexity computed using the subset of the language corpus excluded from the set used for LM construction. Application of an LM prepared in this way in ASR reduced the overall WER of speech recognition by about 4%.

Although many concepts aimed at constructing effective LMs for ASR have been proposed and investigated for other languages, still relatively little work has been carried out towards the verification of these concepts in the case of Polish ASR. The method's properties confirmed in the environment of one language may not be confirmed for other languages. Therefore, the ultimate aim of the works described in this paper is to verify the effectiveness of methods related to LM construction for the sake of ASR in Polish and to combine selected methods into a consistent procedure of LM construction. In the proposed procedure we try to take into account specific features and problems related to Polish ASR. The problems considered here are as follows:

- lack of large corpora published as full texts (*in extenso*)—as a result, language models must be built from limited corpora (this particularly concerns narrow domain ASR applications, e.g., in medical information systems);
- loose word order in sentences;
- frequent appearance of ASR errors typical for Polish phonetics like the tendency for short word deletion;
- practical issues occurring in ASR applications: the necessity to boost recognition accuracy of very important utterances (that may not be represented in the corpus used to build the LM) and insertion of out-of-corpus words into the language model.

The method of LM construction proposed and experimentally evaluated in this article creates the LM by applying a sequence of stages. Each stage modifies or extends the LM provided to its input by applying the operation aimed at the specific LM feature. For this reason we called the proposed method the *pipelined LM construction*. At each stage, various alternative methods are experimentally evaluated. The one that gives the LM which exhibits the highest ASR accuracy in tests is recommended.

The first stage consists in building an initial smoothed stochastic LM. At this stage we compare various smoothing methods, in order to test whether there are significant differences in ASR accuracy between LMs created using various smoothing methods. The best smoothing method is used to build the initial LM. It is then passed to the next stage, where the model is improved by taking into account indirectly co-occurring words. The proposed method boosts the probability of bigrams corresponding to indirectly co-occurring words. The next stage modifies the LM so as to avoid short word deletion errors as much as possible. The fourth stage introduces out-of-corpus words to the LM. These are words that do not appear in the corpus utterances but are necessary in a particular ASR application. At the last stage the LM is modified so as to increase the accuracy of very important phrase recognition. These phrases are utterances (or fragments of utterances) which are of crucial importance for the speaker. They may be under-represented in the corpus, thus their recognition accuracy may be not sufficient.

In pipelined language model construction we try to achieve WER reduction by applying specific methods of model extension or modification. At all stages except for the first one, our own methods were used. At the second stage of the pipeline we used our own method of distant co-occurring bigram boosting, outlined earlier (Sas and Żołnierek, 2011). Here its performance was evaluated in various domain-specific areas of ASR application and compared to that of the LM created using typical smoothing methods. At the third stage, the concept of multi-words was utilized. It is not quite new and was used by other authors. In the approach presented by Chen and Chan (2003) multi-words are used as specific collocation contexts for other words that are frequently associated with them. Kolorenc *et al.* (2006) apply the multi-word concept for the same purpose as we consider here. They, however, create multi-words by only analyzing short word occurrence in specific bigrams in the corpus.

We used here a slightly different, novel approach. The novelty consists in taking into account acoustic similarities as the criterion of multi-word application is the particular context. The idea of combing the LM with the flat out-of-corpus word list used here at the fourth stage of the pipeline can be found in the work of Brown *et al.* (1992). In our work we applied the class-based approach to Polish language modeling, using *Part-Of-Speech* (POS) tagging. n-gram probabilities for classes are estimated in a typical way from the corpus. The method used at this stage is not quite novel because it just combines techniques presented by other authors. Our aim here was to find out if this known approach can significantly improve the accuracy of out-of-corpus word recognition. At the last stage our own method is applied, which modifies the model so as to achieve high accuracy of very important utterance recognition. The element of novelty at this stage consists in using the HMM as the tool

for artificial speech samples generation.

The organization of the paper is as follows. In Section 2 the approach to ASR based on the LM is shortly described. The next section presents selected LM smoothing techniques which were used as candidates to create the baseline LM at the first stage of the pipeline. In Section 4, the method of co-occurring n-gram probabilities boosting is presented in detail. Experimental evaluation and comparison of LMs created using various smoothing techniques in Polish speech recognition are described in Section 5. The idea of application of multi-words to short word deletion error avoidance is presented in Section 6. Section 7 is devoted to the method of combing the LM with the flat word list. In Section 8, the method of boosting the probabilities of very important phrases is described. For all new methods presented in Sections 6–8, the results of their empirical investigations are included in the corresponding sections. The last section presents conclusions and further research directions.

2. Automatic speech recognition with acoustic and language models

A typical approach to the problem of automatic speech recognition consists in building acoustic models and language models combined into a compound hidden Markov model. HMMs proved to be an efficient technique in modeling sequential processes related to various man-machine interactions as speech, handwriting or gesture recognition (Kasprzak *et al.*, 2012). Although other approaches were tested in the ASR domain, the HMM still remains the primary speech modeling and recognition technique.

The HMM speech model can be considered a three-level system. On the lowest level, simple Markov models for individual phonemes specific to the language are created and trained. A uniform HMM topology for each phoneme is assumed. It consists of three observation emitting states. The state transition probabilities as well as the parameters of observations emission probability density functions for all phoneme HMMs are estimated using the Baum–Welch procedure. On the middle level, the models of words are created by concatenating models of subsequent phonemes appearing in the phonetic transcription of the word. Because the phoneme HMMs can be multiplied applied in various words, training of the HMM for the language consisting of a set of words does not require all words from the language to be presented during training. Then for each admissible word from the dictionary $\mathcal{D} = (w_1, w_2, \ldots, w_N)$ we deal with the word HMM which is built by concatenating HMMs for subsequent phonemes.

Finally, on the highest level, the compound HMM of the whole utterance is built by connecting word HMMs in one language HMM. The probabilities of transition from the terminal state of a word HMM to the initial state of another word HMM are taken from the domain-specific n-gram language model. Details of this procedure are presented in the next section. In automatic speech recognition we start with acquisition of the speech acoustic signal from the sound device and segment it into fragments being individual utterances separated by silence. The isolated utterances are recognized independently. Every utterance being recognized is converted into a sequence of vectors of observations $(o_1, o_2, \ldots, o_t)$. Then, finally, the recognition with a compound HMM consists in finding such a word sequence W^* which maximizes its conditional probability given the sequence of observations:

$$W^* = \arg \max_{w_{i_1},\ldots,w_{i_k} \in \mathcal{D}^+} P(w_{i_1}, \ldots, w_{i_k} | o_1, o_2, \ldots, o_t)), \tag{1}$$

where $\mathcal{D}^+$ denotes the set of all nonempty sequences of words from the dictionary $\mathcal{D}$.

3. Backoff LM smoothing

We will be considering here the languages being sets of sequences of words coming from the finite dictionary $\mathcal{D}$. A stochastic n-gram language model is the set of data that makes it possible to estimate the probability of appearance of the n-th word w_i provided that the sequence of preceding $n-1$ words $w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1}$ is known. In other words, the LM provides the method to compute the estimation of

$$p(w_i | w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1}). \tag{2}$$

A sequence of n consecutive words is called the n-gram. The conditional probabilities can be given explicitly in the LM or they can be defined procedurally. The language model is usually constructed from the language corpus which is a sufficiently large set of sample phrases in the language being modeled. The most obvious way to find out the probability estimates is to count the occurrences of n-grams in the model and to apply *Maximum Likelihood* (ML) estimation:

$$p_{ML}(w_i | w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1}) = \frac{c(w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1}, w_i)}{c(w_{i-n+1}, w_{i-n+2}, \ldots, w_{i-1})}, \tag{3}$$

where $c(w_1, w_2, \ldots, w_n)$ is the number of n-gram occurrences $w_1, w_2, \ldots, w_n$ in the corpus. Due to a limited size of the corpus, nonzero ML estimates can be obtained only for a very limited set of n-grams. For this reason, in practice, low n-gram orders are used—in most cases n does not exceed 3.

In our experiments we use a bigram LM, which corresponds to setting $n = 2$. Limiting the n-gram order still does not solve the data sparseness problem completely in the case of languages consisting of thousands of words. It is still very likely that many n-grams that may appear in typical language use are missing in the corpus or the number of their occurrences is not big enough to allow reliable ML estimation of related probabilities. To prevent underestimation of probabilities of missing n-grams, the concept of a *backoff* is applied. Backing off consists in using lower order n-gram probabilities when the number of occurrences of an n-gram in the corpus is not sufficient. In such a case the probability (2) is approximated using the lower order n-gram probability $p(w_i|w_{i-n+2}, \ldots, w_{i-1})$.

In the bigram LM the probabilities $p(w_i|w_{i-1})$ are approximated by prior w_i word probabilities $p(w_i)$ which in most cases can be reliably estimated with ML estimators:

$$p(w_i) = \frac{c(w_i)}{\sum_{w \in \mathcal{D}} c(w)}, \tag{4}$$

where $c(w)$ is the number of unigram w occurrences in the corpus. While missing n-gram probabilities estimated with the ML estimator are underestimated (nulled), the probabilities of n-grams occurring in the corpus only a few times are usually over-estimated. Therefore the concept of the backoff is complemented with that of *discounting*. Probability discounting consists in subtracting some probability mass from the probabilities (2) estimated with the ML based on the formula (3). As result, for bigrams[1] occurring in the corpus, the probability $p_{ML}(w_i|w_{i-1})$ estimated using an ordinary ML estimator is replaced by the discounted probability $p_d(w_i|w_{i-1}) \leq p_{ML}(w_i|w_{i-1})$.

In discounted backoff LMs, the probability mass discounted from the ML estimates of probabilities $p_{ML}(w_i|w_{i-1})$ for bigrams actually occurring in the corpus is distributed among words that never occurred as successors of w_{i-1}. The discounted probability mass $\beta(w)$ for any word w can be computed as

$$\beta(w) = 1 - \sum_{w_k : c(w,w_k) > 0} p_d(w_k|w), \tag{5}$$

where $c(w_i, w_{i-1})$ is the number of bigram (w_i, w_{i-1}) occurrences in the corpus. The conditional probabilities of words w_i that never appeared as successors of w are proportional to their prior probabilities computed according to (4). The probabilities $p(w_i|w)$, however, have to sum up to 1.0 for every word w over all words from the dictionary $\mathcal{D}$. Therefore the probabilities for bigrams (w, w_i) not observed in the corpus are finally computed as

$$p(w_i|w) = \alpha(w) p(w_i), \tag{6}$$

where

$$\begin{aligned} \alpha(w) &= \frac{\beta(w)}{\sum_{w_k : c(w,w_k)=0} p(w_k)} \\ &= \frac{\beta(w)}{1 - \sum_{w_k : c(w,w_k)>0} p(w_k)}. \end{aligned} \tag{7}$$

Various schemes of discounting were proposed and tested in various LM applications (Goodman, 2001). In our test, for the comparison purpose, the following smoothing methods were chosen:

- Good–Turing (GT) estimate,
- Absolute Discounting (AD),
- Kneser–Ney (KN) smoothing,
- Modified Kneser–Ney (MKN) smoothing.

The details of the methods can be found in the work of Goodman (2001), and their effectiveness in Polish ASR will be investigated in Section 5. Let us briefly recall the ideas of above mentioned methods, using the notation of Goodman (2001).

The **Good–Turing estimate** states that for any bigram that occurs r times we should pretend that it occurs r^* times:

$$r^* = (r+1)\frac{n_{r+1}}{n_r}, \tag{8}$$

where n_r is the number of bigrams that occur exactly r times in the corpus. Consequently, we can calculate the probability for a bi-gram δ with r counts,

$$p_{GT}(\delta) = \frac{r^*}{N}, \tag{9}$$

where $N = \sum_{r=0}^{\infty} n_r r^*$.

In **absolute discounting**, the bigram probability estimate $p_{ABS}(w_i|w_{i-1})$ is computed by subtracting a fixed discount $0 \leq d \leq 1$ from each nonzero count of bigram occurrences, and by mixing the discounted ML bigram estimate with the unigram estimate of the successor word w_i, i.e.,

$$\begin{aligned} p_{ABS}(w_i|w_{i-1}) = &\frac{\max[c(w_{i-1}, w_i) - d, 0]}{\sum_{w \in \mathcal{D}} c(w_{i-1}, w)} + \\ &+ (1 - \lambda_{w_{i-1}}) p_{ABS}(w_i). \end{aligned} \tag{10}$$

To make this distribution sum up to 1, we should take

$$1 - \lambda_{w_{i-1}} = \frac{d}{\sum_{w \in \mathcal{D}} c(w_{i-1}, w)} N_{1+}(w_{i-1}\bullet), \tag{11}$$

where the number of unique words that follow the predecessor w_{i-1} is defined as

[1] In the further part of the paper we will restrict our discussion to bigram language models.

$$N_{1+}(w_{i-1}\bullet) = \mid \{w_i : c(w_{i-1}, w_i) > 0\} \mid . \quad (12)$$

The notation N_{1+} is meant to evoke the number of words that have one or more counts, and $\bullet$ is meant to evoke a free variable that is summed over.

Kneser–Ney smoothing is an extension of absolute discounting where the lower-order distribution that one combines with a higher-order distribution is built in another manner. For a bigram model, we select a smoothed distribution p_{KN} that satisfies the following constraint on unigram marginals for all w_i:

$$\sum_{w_{i-1}\in\mathcal{D}} p_{KN}(w_{i-1}, w_i) = \frac{c(w_i)}{\sum_{w\in\mathcal{D}} c(w)}. \quad (13)$$

The Kneser–Ney model can be presented in the same recursive way as (10), i.e.,

$$p_{KN}(w_i|w_{i-1}) = \frac{\max[c(w_{i-1}, w_i) - d, 0]}{\sum_{w\in\mathcal{D}} c(w_{i-1}, w)} + \kappa(w_{i-1}) p_{KN}(w_i). \quad (14)$$

where

$$\kappa(w_{i-1}) = \frac{d}{\sum_{w\in\mathcal{D}} c(w_{i-1}, w)} N_{1+}(w_{i-1}\bullet). \quad (15)$$

The unigram probabilities can be calculated as follows:

$$p_{KN}(w_i) = \frac{N_{1+}(\bullet w_i)}{N_{1+}(\bullet\bullet)}, \quad (16)$$

where

$$N_{1+}(\bullet w_i) = \mid \{w_{i-1} : c(w_{i-1}, w_i) > 0\} \mid \quad (17)$$

and

$$N_{1+}(\bullet\bullet) = \sum_{w_i\in\mathcal{D}} (N_{1+}(\bullet w_i)). \quad (18)$$

In **modified Kneser–Ney smoothing**, the method proposed by Goodman (2001), instead of using a single discount d for all nonzero counts we use three different parameters d_1, d_2 and d_{3+} that are applied to bigrams with one, two, and three or more counts, respectively. Now the formula (14) turns into

$$p_{MKN}(w_i|w_{i-1}) = \frac{c(w_{i-1}, w_i) - d(c(w_{i-1}, w_i))}{\sum_{w\in\mathcal{D}} c(w_{i-1}, w)} + \gamma(w_{i-1}) p_{MKN}(w_i) \quad (19)$$

where

$$d(c) = \begin{cases} 0 & \text{if } c = 0, \\ d_1 & \text{if } c = 1, \\ d_2 & \text{if } c = 2, \\ d_{3+} & \text{if } c \geq 3. \end{cases} \quad (20)$$

To make the distribution sum to 1, we take

$$\gamma(w_{i-1}) = \frac{d_1 N_1(w_{i-1}\bullet) + d_2 N_2(w_{i-1}\bullet) + d_{3+} N_{3+}(w_{i-1}\bullet)}{\sum_{w\in\mathcal{D}} c(w_{i-1}, w)}, \quad (21)$$

where $N_2(w_{i-1}\bullet)$ and $N_{3+}(w_{i-1}\bullet)$ are defined analogously to $N_1(w_{i-1}\bullet)$ (12).

4. Probability boosting for indirectly co-occurring n-grams

The idea presented by Sas and Żołnierek (2011) lies in increasing the probability $p(w_k|w)$ for words w and w_k that occur in the corpus close each to other but do not necessarily appear in adjacent positions. The idea behind this concept is motivated by the fact that, in loose word order languages like Polish, if two words co-occur in the same utterance then it is likely that they will occur in other utterances in adjacent positions. Thus, appearance of the co-occurrence of words in the corpus in distant positions can be an indication to increase the probability of the corresponding bigram. We assume here that the language corpus consists of clearly separated utterances. The probability of a bigram will be boosted if its two components co-occur in the same utterance.

Let us consider a single word $w \in \mathcal{D}$. Let $\mathcal{N}(w)$ denote the set of words that appear at least once in the corpus directly after the word w, i.e., $\forall w_i \in \mathcal{N}(w) : c(w, w_i) > 0$. By $\mathcal{F}(w)$ we will denote the set of words that co-occur in at least one utterance with the word w but do not belong to $\mathcal{N}(w)$. $\mathcal{X}(w)$ denotes all remaining words from the dictionary ($\mathcal{X}(w) = \mathcal{D} \setminus \mathcal{N}(w) \setminus \mathcal{F}(w)$).

In the ordinary LM the probabilities for bigrams (w, w_i) for $w_i \in \mathcal{F}(w) \cup \mathcal{X}(w)$ are calculated according to (6), where $\alpha(w)$ is uniformly calculated using the formula (7). In order to boost the probability of bigrams consisting of words from $\mathcal{F}(w)$, their probabilities will be increased by the factor $\lambda > 1.0$, common for the whole model, i.e.,

$$\forall w_i \in \mathcal{F}(w) : p(w_i|w) = \lambda\alpha(w) p(w_i). \quad (22)$$

We assume that the total discounted probability mass $\beta(w)$ defined in Eqn. (5) remains unchanged. To achieve this, the probabilities assigned to bigrams consisting of words from $\mathcal{X}(w)$ must be lowered appropriately. Now the probabilities (6) are multiplied by the factor $\bar{\lambda}(w) < 1.0$, which must be individually calculated for each w, so as the total probability mass $\beta(w)$ is preserved:

$$\lambda\alpha(w) \sum_{w_k\in\mathcal{F}(w)} p(w_k) + \bar{\lambda}(w)\alpha(w) \sum_{w_k\in\mathcal{X}(w)} p(w_k) = 1 - \sum_{w_k\in\mathcal{N}(w)} p_d(w_k|w) = \beta(w), \quad (23)$$

where $p_d(w_k|w)$ is the discounted probability obtained with any discounting method. Hence, $\bar{\lambda}(w)$ can be calculated as

$$\bar{\lambda}(w) = \frac{1 - \sum\limits_{w_k \in \mathcal{N}(w)} p_d(w_k|w) - \lambda\alpha(w) \sum\limits_{w_k \in \mathcal{F}(w)} p(w_k)}{\alpha(w) \sum\limits_{w_k \in \mathcal{X}(w)} p(w_k)}. \tag{24}$$

Finally, the probability $p^*(w_i|w)$ that the modified language model assigns to a bigram (w, w_i) can be defined as

$$p^*(w_i|w) = \begin{cases} p_d(w_i|w) & \text{if } w_i \in \mathcal{N}(w), \\ \lambda\alpha(w)p(w_i) & \text{if } w_i \in \mathcal{F}(w), \\ \bar{\lambda}(w)\alpha(w)p(w_i) & \text{if } w_i \in \mathcal{X}(w). \end{cases} \tag{25}$$

The value of λ, common for the whole model, can be computed so as to maximize the probability that the model assigns to the separated fragment of the language corpus. In order to do it, the corpus is divided into two parts: the training part $\mathcal{T}$ and evaluation part $\mathcal{E}$. The latter is a set of word sequences $s_1, s_2, \ldots, s_n$. Assume that each sequence starts with the specific start tag "<s>" and ends with the end tag "</s>":

$$s_j = (< \mathtt{s} >, \mathtt{w}_1^{(\mathtt{j})}, \mathtt{w}_2^{(\mathtt{j})}, \ldots, \mathtt{w}_{\mathtt{l(s_j)}}^{(\mathtt{j})}, < /\mathtt{s} >). \tag{26}$$

The probability that the LM assigns to the utterance s_j can be computed as

$$P(s_j; LM(\mathcal{T}, \lambda)) = \prod_{k=1}^{l(s_j)+1} p^*(w_k^{(j)}|w_{k-1}^{(j)}; LM(\mathcal{T}, \lambda)), \tag{27}$$

where $w_0^{(j)} = < \mathtt{s} >$, $w_{l(s_j)+1}^{(j)} = < /\mathtt{s} >$ and $l(s_j)$ denotes the length of the utterance s_j. The probability that the language model $LM(\mathcal{T}, \lambda)$ assigns to the whole evaluation set $\mathcal{E}$ is

$$P(\mathcal{E}; LM(\mathcal{T}, \lambda)) = \prod_{s \in \mathcal{E}} P(s; LM(\mathcal{T}, \lambda)). \tag{28}$$

The value of λ is determined in an iterative procedure so as to maximize the probability (28).

5. Experimental evaluation of LMs in Polish speech recognition

In order to evaluate the performance of LMs created using the methods described in the previous sections a series of experiments was carried out. The LM performance is assessed by (a) the perplexity computed on the sentence set representative for a domain and (b) the word error rate of a speech recognizer which uses the LM being evaluated. The perplexity is the measure of LM quality. It is based on the average probability that the tested LM assigns to sentences in the test set computed "per word" (Goodman, 2001). The lower the LM perplexity, the better the language stochastic properties approximation by the LM. However, from the practical point of view the ultimate LM assessment should be rather determined by evaluating the LM contribution to the accuracy increase of the ASR process.

Four domains of the Polish language which differ in complexity were used in the experiment[2]:

- CT: texts from the domain of medical diagnostic image reporting, mainly related to CT and MRI modalities; dictionary size—23 thousands of words, corpus size—22 MB;
- TL: a collection of not copyrighted texts of the Polish literature or foreign language books translated to Polish; dictionary size—81 thousands of words, corpus size—8 MB;
- GM: general medicine texts consisting of elements of medical documentation, medical examination reports and medical articles collected from Wikipedia; dictionary size—119 thousands of words, corpus size—94 MB;
- PL: general purpose Polish language texts consisting of the Polish literature (8%), medical documentation samples (11%), samples of newspaper articles (1%), reports from the Polish Parliament sessions (35%), Senate of Republic of Poland proceedings (35%), European Parliament Proceedings from Polish-English parallel corpus (Koehn, 2005) (10%); dictionary size—576 thousands of words, corpus size—370 MB.

The experiment consists of two stages. At the first stage, the performance of typical language models built with the smoothing techniques described in Section 3 are compared by their perplexities and by the WER of a speech recognizer based on the model being compared. Then the best smoothing method, taking into account the WER, is selected for further experiments. At the second stage, the chosen method is combined with the backoff technique proposed here. The speech recognition accuracy obtained using the modified model is compared with the corresponding accuracy achieved with the conventional LM.

The acoustic model for ASR was created in a speaker-dependent manner as a triphone model using speech samples recorded by a single male speaker. The

[2]The corpora used to build the LMs employed in the experiments described in this article as well as the recorded utterances used in the acoustic models are available at http://sun10.ci.pwr.wroc.pl/~sas/ASR.

total duration of training utterances is about 5 hours. For ASR accuracy testing the individual set of utterances was used for each domain LM. The duration of the utterances in the test set was in the range of 50–70 minutes. The HTK toolkit (Young and Everman, 2009) was applied to build the acoustic model. The open source recognition engine Julius (Lee *et al.*, 2001) was used as the speech recognizer.

The results of the first stage of the experiment concerning perplexity comparison and ASR accuracy evaluation are shown in Tables 1 and 2, respectively. Just for the sake of comparison, results obtained with the unigram (U) LM are also shown. No smoothing is applied in the case of the unigram LM. The symbols AD, GT, KN, MKN denote methods of smoothing as described in Section 3, while CT, TL, GM, PL denote the language corpora applied. Additionally, Table 2 contains the column "Total", where the ASR accuracy of the combination of utterances from the specific domains CT, TL, GM, PL is presented. For the sake of statistical significance evaluation of the obtained results, the confidence interval radius ϵ was determined for each smoothing method and the utterance combination. The confidence level $1 - \alpha = 0.9$ was used.

It can be observed that there are practically no significant differences between bigram models created with various discounting methods. Despite the loose word order of the Polish language, bigram models have much lower perplexity than unigram models (U). The bigram/unigram perplexity ratios for Polish are similar to these reported for English by Jurafsky and Matrin (2009). ASR in Polish is, however, much less sensitive to LM perplexity than English. Goodman (2001) claims that the increase of cross-entropy (which is a logarithm of perplexity) by 0.2 results in the absolute increase of the WER by 1%. The results presented in Tables 1 and 2 show much weaker dependence of the WER on the model cross-entropy. Let us consider cross-entropies and WERs of the CT and PL models obtained using the KN method. According to Goodman (2001), cross-entropy H is defined as $H(T) = \log_2(PP(T))$, where PP is the perplexity and T is the test set. The difference of the cross-entropy between CT and PL models is $\log_2(633.3) - \log_2(34.4) = 4.2$.

The increase in cross-entropy by 4.2 results in the increase of the WER by only about 3.5%. Also the superiority of the modified Kneser–Ney smoothing method reported by Goodman (2001) over all other methods is not confirmed in the case of the Polish language. The performance of the modified Kneser–Ney model is even slightly worse than that of the original Kneser–Ney model. Although the performances of all bigram models are similar, the original Kneser–Ney model achieves the best results both in perplexity and WER comparisons. To verify the statistical significance of the obtained result, the confidence intervals of ASR accuracy estimation were determined at the confidence level $1 - \alpha = 0.9$. The analysis of confidence intervals presented in the rightmost column in Table 2 shows that only the unmodified Kneser–Ney method marginally outperforms other compared techniques. It was used for further experiments.

At the second stage, Kneser–Ney smoothing was combined with our bigram boosting technique described in Section 4. The accuracy of the speech recognizer was then evaluated for the modified model. ASR accuracies in various language domains are presented in Table 3. CT, TL, GM and PL denote language corpora applied. The rightmost column contains combined results obtained using the mixture of utterances coming from individual domains. The results of the Kneser–Ney method (the same as in Table 2) used here as a baseline for comparison are presented in the first row. The accuracies obtained using the LM created with the bigram boosting method from Section 4 are presented in the second row. The last row contains the achieved relative WER reduction rate.

The application of indirectly co-occurring bigrams boosting resulted in small but observable improvement of the ASR accuracy in the case of all language domains except for the simplest CT model. The average relative WER reduction is about 5.7%. To show the statistical significance of the obtained results, the confidence interval radiuses of the accuracy estimates were calculated for the combined test utterances set. The radii for accuracy estimaties for baseline Kneser–Ney and the

Table 1. Perplexity of models based on various smoothing methods (the first row contains results obtained with the unsmoothed unigram model).

Method	CT	TL	GM	PL
U	751.5	2688.3	1596.3	4233.6
AD	35.9	748.4	69.2	672.1
GT	35.9	733.2	68.2	665.4
KN	34.5	713.0	65.7	633.3
MKN	34.9	720.4	66.6	633.5

Table 2. ASR accuracy obtained with models based on various smoothing methods (ϵ: confidence interval radius).

Method	CT	TL	GM	PL	Total
U	0.937	0.893	0.925	0.904	0.914 ϵ=0.0025
AD	0.953	0.906	0.942	0.916	0.928 ϵ=0.0023
GT	0.952	0.908	0.949	0.920	0.932 ϵ=0.0022
KN	0.960	0.914	0.951	0.924	0.936 ϵ=0.0022
MKN	0.957	0.917	0.949	0.920	0.934 ϵ=0.0022

bigram boosting method are shown in the rightmost column of Table 3. The confidence intervals do not overlap, hence the superiority of the model created using bigram boosting seems to be statistically significant.

6. Application of multi-words

Some words appear most frequently in specific collocations. Pairs of strongly collocated words can be replaced in the LM by their combinations called here *multi-words*. Application of multi-words extends the context represented in the n-gram LM and hence makes it possible to model a language more precisely without extension of its order. On the other hand, however, too intensive application of multi-words reduces LM generalization abilities because it assigns more probability to n-grams actually appearing in the corpus and lowers the backoff probability mass. In the extreme case, if all sentences appearing in the text corpus are converted to multi-words, only utterances represented in the corpus can be recognized. The problem is therefore which word pairs appearing in the corpus should be replaced by the corresponding multi-words.

Application of multi-words can be particularly beneficial in the case of two-word collocations where at least one word is very short (in particular, where it consists just of a single phoneme). Our experiments with ASR applied to Polish show that recognizers exhibit strong tendency to discard *Short Words* (SW) from the recognized phrase. It is most often observed if the phone ending the preceding word or beginning the next word is acoustically similar to the phone being the pronunciation of the short word. We will call such the situation the *deleting context* and resulting recognition error will be called the *Short Word Deletion* (SWD) error, for short. Experiments with ASR in Polish (Sas, 2010) show that ASR decoders tend to falsely skip a short word in deleting contexts, which leads to approximately 3% of the WER. Combining a short word with the adjacent one could improve ASR accuracy, in particular in the case when word insertion penalties are used.

The concept of multi-words is not new and its application to improve LM prediction abilities is described in the literature. In the approach presented by Chen and Chan (2003) multi-words are used as a specific collocation contexts for other words that are frequently associated with them. Kolorenc *et al.* (2006) apply the multi-word concept for the same purpose as we consider here. They, however, create multi-words by only analyzing short word occurrence in specific bigrams in the corpus. The novelty of the approach presented here lies in taking into account acoustic similarities as the criterion of multi-word selection.

Table 3. ASR accuracy and WER reduction obtained with the model based on indirectly co-occurring bigram probability boosting (ϵ: confidence interval radius).

	CT	TL	GM	PL	Total
Kneser–Ney(KN)	0.960	0.914	0.951	0.924	0.936 ϵ=0.0022
Bigrams boosting	0.959	0.920	0.955	0.929	0.940 ϵ=0.0021
Relative WER reduction	-2.5%	6.9%	8.1%	6.5%	5.7%

We compared two approaches aimed at SWD avoidance. The first one utilizes the multi-word concept. It consists in text corpus modification by concatenating words constituting deletion contexts into multi-words. Then the typical language model building procedure is applied to the modified corpus. If a multi-word is recognized by the decoder, it is split into its component words at the post processing stage executed after the basic recognition of an utterance.

The second approach consists in boosting bigram probabilities for bigrams corresponding to deletion contexts in a way similar to the one described in Section 4. Both methods lead to LM modification oriented to SWD error rate decrease.

The first problem that needs to be solved is how to select pairs of words that constitute the deletion context. Our experiences with ASR applied to the Polish language proved that deleting contexts comprise mainly pairs of words where one of them is a single-phoneme word. In Polish language there exist the following single phoneme words: *'i'* (and), *'a'* (and/but depending on the context), *'o'* (about), *'u'* (at), *'w'* (in/inside), *'z'* (with/out of). The strongest tendency to delete the short word appears if the neighboring phoneme in the adjacent word is the same as the phoneme being the short word. A weaker tendency of SWD errors appears in the situations where *'w'* or *'z'* are pronounced as voiceless consonants and the neighboring phoneme in adjacent word is also voiceless.

Experiments described by Sas (2010) show that most of SWD errors are related to *'w'* and *'z'* preposition deletion in deleting contexts. In Polish single-word prepositions and conjunctions causing a majority SWD errors are most strongly collocated with successive words. Therefore we will restrict our discussion to deleting contexts where the short word is the first element of a bigram. In the text corpus related to medical diagnostic image reporting used in experiments, the relative frequency of *'w'*/*'z'* preposition occurrence in deleting contexts is about 3%. With the average SWD probability in deleting contexts close to 0.5, this introduces the overall deletion error rate 1.5%. With the overall error rate achievable for speaker dependent ASR close to 8%, SWD errors constitute about 20% of word errors in speech recognition. Reduction of

the SWD error rate would then observably contribute to speech recognition accuracy improvement. In the method proposed here we focus on typical deleting contexts that are represented by bigrams appearing in the text corpus used to create the language model; however, the concepts applied can be extended also to other deleting contexts for bigrams not appearing in the corpus.

6.1. Application of multi-words to reduction of SWD errors. This method of reducing SWD errors consists in a selective replacement of bigrams corresponding to deleting contexts by multi-words being a concatenation of original words constituting the deletion context. In order to preserve the generalization ability of the resultant LM, not all but only randomly selected deleting contexts are replaced by multi-words. The probability of the replacement of a particular occurrence of the deleting context intuitively should depend on the SWD probability in this context. Alternatively, it can be set up so as to obtain the best model according to the cross-perplexity criterion. The replacement probabilities are defined for distinguished groups of deleting contexts. The groups are distinguished based on acoustic similarities of adjacent phonemes:

- A: SW is *'w'* and the right neighboring word pronounced in this context begins with the same phoneme;
- B: SW is *'w'* and the right neighboring word pronounced in this context begins with a voiceless phoneme, e.g. *'w sytuacji'* (*'in the situation...'*), *'w przypadku'* (*'in the case ...'*);
- C: SW is *'z'* and the right neighboring word pronounced in this context begins with the same phoneme;
- D: SW is *'z'* and the right neighboring word pronounced in this context begins with a voiceless phoneme, e.g., *'z samym'* (*'with only...'*), *'z powodu'* (*'because of ...'*);
- E: SW is *'a'*, *'i'*, *'o'* or *'u'* and the right neighboring word pronounced in this context begins with the same phoneme.

Let p_A, p_B, p_C, p_D, p_E denote the probabilities of the replacement of deleting contexts by the corresponding multi-words in the text corpus. Two methods were considered to determine these probabilities:

- the probabilities p_X are just estimates of SWD errors made by the speech recognizer in corresponding groups of deleting contexts X—the higher the SWD probability in a group, the more frequently the deleting context is replaced by the multi-word in the corpus;
- the probabilities are determined so as to maximize the cross perplexity of the obtained model tested on the evaluation text set Ω disjoint form the corpus used to create the language model.

The first method utilizes the actual tendency of the speech recognizer to SWD errors, but unfortunately makes the resultant LM dependent on acoustic properties of the speech and hence introduces speaker dependency. An LM optimized for one speaker may be inappropriate for another.

The second method is independent of the speaker. Let $LM(p_A, \ldots, p_E)$ denote the model created with the modified corpus where the probabilities of deleting context replacement in groups $A, \ldots, E$ are $p_A, \ldots, p_E$. Let $w_{i,1}, w_{i,2}, \ldots, w_{i,l(i)}$ denote the sequence of words constituting the i-th utterance in the evaluation set. Let $m_{i,1}, m_{i,2}, \ldots, m_{i,l_M(i)}$ denote the sequence corresponding to the i-th utterance, where word pairs constituting deleting contexts were replaced by corresponding multi-words ($m_{i,j}$ is here either the original word occurring in the utterance or a multi-word obtained by deleting context substitution). The numbers $l(i)$ and $l_M(i)$ denote lengths of the original and modified i-th utterance.

The average probability per word that the LM assigns to the i-th utterance $w_{i,1}, w_{i,2}, \ldots, w_{i,l(i)}$ can be calculated as follows:

$$\begin{aligned}&\widetilde{p}(w_{i,1}, w_{i,2}, \ldots, w_{i,l(i)}; LM(p_A, \ldots, p_E))\\ &= \max\{(\prod_{j=1}^{l(i)} p(w_{i,j}|w_{i,j-1}))^{1/l(i)},\\ &\quad (\prod_{j=1}^{l_M(i)} p(m_{i,j}|m_{i,j-1}))^{1/l_M(i)}\}.\end{aligned} \tag{29}$$

Because both variants (consisting of original words or containing corresponding multi-words) of the utterance are equally accepted, the greater of the two probabilities $p(w_{i,1}, w_{i,2}, \ldots, w_{i,l(i)})$ and $p(m_{i,1}, m_{i,2}, \ldots, m_{i,l_M(i)})$ is selected. The probability assigned to the whole evaluation set consisting of n_Ω utterances is calculated as a product of probabilities defined in Eqn. (29):

$$\begin{aligned}&p(\Omega; LM(p_A, \ldots, p_E))\\ &= \prod_{i=1}^{n_\Omega} \widetilde{p}(w_{i,1}, w_{i,2}, \ldots, w_{i,l(i)}; LM(p_A, \ldots, p_E)).\end{aligned} \tag{30}$$

The probabilities $p_A, \ldots, p_E$ should be set so as to maximize the probability (30).

6.2. Reduction of SWD errors by direct modifications of probabilities in the LM. An alternative method consists in direct boosting of bigram probabilities in

the LM. Now we assume that we start with the LM created using the methods described in Section 4. Our aim is to modify the LM so as to reduce the total WER by limiting the rate of SWD errors. Increasing probabilities $p(w_{i-1}|w_i)$ for word pairs (w_{i-1}, w_i) constituting deleting contexts obviously leads to SWD error reduction, but it may also increase the rate of errors of false insertions of short words in contexts where they actually do not occur. Therefore the method of conditional probability modification in the LM should keep balance between resultant tendencies to reduce SWD and false short word insertion errors.

The idea of the approach proposed here is similar to the one presented by Sas (2010). The method described in this article assumes a kind of post processing, therefore it may not be applicable to standard ASR tools, which do not make it possible to change its processing pipeline. Here we used the modification concerning only the LM.

As has already been pointed out, in Polish most of single-phoneme words are strongly collocated rather with the successive word. Our aim is therefore to boost the probability $p(w_{i-1}|w_i)$ for words constituting deleting contexts, where w_{i-1} is a short word. Making a decision on the boost rate individually for deleting contexts is infeasible due to a lack of specific information about the recognizer tendency to make SWD errors in individual contexts. It seems rather reasonable to apply the same boost rate for deleting contexts that are acoustically similar, which results in setting the boost rate for groups $A, \ldots, E$ specified in the previous section. One way to achieve the probability boost is to apply the power function to the original probability:

$$p'(w_{i-1}|w_i) = p^{\mu_X}(w_{i-1}|w_i), \quad X \in \{A, \ldots, E\}, \tag{31}$$

where $0 < \mu_X < 1$. The μ_X factors are established for deleting contexts belonging to groups $A, \ldots, E$.

The typical LM bigram is a "forward" model, i.e., it computes the unigram (prior) probabilities of words and conditional probabilities of a successive word conditioned on its predecessor. The proposed methods explicitly modifies backward probabilities $p(w_{i-1}|w_i)$. Then the modified probabilities need to be converted to forward probabilities $p'(w_i|w_{i-1})$ contained in the typical forward model. The conversion can be obtained by simple application of the Bayes rule:

$$p'(w_i|w_{i-1}) = p'(w_{i-1}|w_i)\frac{p(w_i)}{p(w_{i-1})}, \tag{32}$$

where the prior probabilities $p(w_i), p(w_{i-1})$ can be taken from the input LM.

Boosting the selected (for the words considered belonging to the deleting contexts, i.e., to the groups $A, \ldots, E$) probabilities according to the formula (31) in effect causes the boosting of the probabilities (32) included in the LM. Only the probabilities of bigrams constituting deleting contexts which appear in the corpus are modified. Consequently, for each word w from the set (*'w'*, *'z'*, *'a'*, *'i'*, *'o'*, *'u'*) the discounted probability mass $\beta(w)$ needs to be updated. Now the formula (5) is modified so as to use $p'_d(w_k|w)$ instead of $p_d(w_k|w)$. For example, for the word $w =$ *'w'* the modified formula is

$$\begin{aligned}\beta(w) =& 1 - \sum_{w_k:(w,w_k)\in(A\cup B)} p'_d(w_k|w) \\ &- \sum_{w_k:(w,w_k)\notin(A\cup B)} p_d(w_k|w),\end{aligned} \tag{33}$$

where A and B are sets of words constituting deleting contexts of *'w'* defined in Section 6.1. Similarly, we can calculate the discounted probability mass $\beta(w)$ for other considered words appearing in deleting contexts. These probability masses, which for every analysed word are less than previously, now are used for calculating backed-off probabilities for bigrams not explicitly represented in the initial model, i.e., using the formulas (6), (7), (22) and (24). Finally, we obtain the modified consistent LM as presented in the formula (25). The complete procedure of LM modification consists of three steps:

1. estimate the boosting factors μ_x for deleting context groups $A, \ldots, E$;
2. recalculate forward conditional probabilities for all pairs of words constituting deleting contexts that are explicitly represented in the LM;
3. update related model parameters $\beta(w)$ to preserve model consistency.

6.3. Experimental evaluation. The performance of described methods was compared experimentally. We also compared it with the results obtained for an alternative idea described by Sas (2010), where SWD errors were corrected at the postprocessing stage carried out after the typical HMM-based recognition process was completed. In order to evaluate the proposed methods, the SWD error rate was estimated using the test set. The test utterances are selected so as to contain at least one deleting context or the word that appears in the corpus as the right neighbor in frequently occurring deleting contexts. The environment for the experiment as well as domain specific text corpora were the same as described in Section 5.

For each domain, the test set was extracted from the corpus and left aside. For the multi-word based method the corpus was appropriately modified and then the LM was created using Kneser–Ney smoothing combined with indirectly co-occurring bigram boosting. For the method consisting in direct modifications of bigram probabilities, the same technique of initial LM creation was used. Three

methods were compared: (a) the one described by Sas (2010), which carries out SWD error correction as the post processing step, (b) the one based on multi-words, where these probabilities were determined using the perplexity minimization, and (c) the one that applies direct bigram probability boosting for deleting contexts. For each of the domains presented in Section 5, the SWD error count was evaluated for the original LM model and for that obtained by applying approaches being compared (further denoted correspondingly by LM_a, LM_b and LM_c). Because deleting context bigram probability boosting may lead to errors consisting in insertion of unuttered short words, false insertion errors were also counted. The performance of the method using the model LM_X in relation to the baseline (unmodified) model LM can be assessed by the gain factor $\eta(LM_X)$ computed as

$$\eta(LM_X) = 1 - \frac{n_d(LM_X) + n_f(LM_X)}{n_d(LM) + n_f(LM)}, \qquad (34)$$

where $n_d(LM)$ and $n_f(LM)$ are counts of SWD errors and false insertion errors occurring in recognition based on the model LM.

The higher (closer to 1.0) the gain factor value, the more effective the model LM_X in relation to the baseline model LM. The assessment results for the methods are presented in Table 4. The results in the columns CT, TL, GM, PL were obtained using utterances coming from corresponding domains specified in Section 5. The rightmost column (Total) contains results computed using the combination of utterances coming from all domains (CT, TL, GM, PL).

For the sake of statistical significance evaluation of the obtained results, SWD error estimates $e(LM_X)$ and their confidence interval radii $\epsilon(LM_X)$ were computed for the compared methods at the confidence level $1-\alpha = 0.9$. The meaning of symbols used in Table 4 is as follows: n_w is the total number of words in the test set, n_{DC} is the the number of deleting contexts in the test set, $e(LM)$, $\epsilon(LM)$ is the estimated SWD error rate in the baseline reference model LM and its confidence interval radius, $e(LM_X)$, $\epsilon(LM_X)$ are the estimated SWD error and its confidence interval radius obtained using the language model LM_X, $\eta(LM_X)$ is the gain factor of the model LM_X with respect to the reference model.

The method consisting in direct modification of deleting context bigram probabilities in the language model results in the highest gain factor. It reduces almost 44% of SWD-related errors. The method based on multi-words application and the substitution probability computed with merely resultant model perplexity minimization exhibits the worst performance. However, with almost 35% of SWD-error reduction, it seems usable in practice as well. The confidence intervals of the estimated SWD errors $(e(LM_X) - \epsilon(LM_X), e(LM_X) + \epsilon(LM_X))$ for the methods being compared do not overlap with the confidence interval of the estimated error of the reference model $(e(LM) - \epsilon(LM), e(LM) + \epsilon(LM))$. Hence the performance of all compared methods related to SWD errors is significantly better than that of the baseline model.

The results presented here were obtained by merely modifying the LM used by the speech recognizer. They can be compared to another alternative method also aimed at SWD-error reduction described by Sas (2010). His method applies an additional postprocessing stage, therefore it is more troublesome in application, in particular, as far as using standard ASR tools is considered.

The average SDW reduction obtained with the alternative method is 0.45, which is a slightly better result than those achieved by methods described in this article. The difference, however, is not practically significant. The superiority of the approach presented here lies in its easier implementation in the case of applying standard ARS tools. Additionally, in the case of LM_a, it can be created in a purely speaker-independent manner without considering any properties of the acoustic model, which is not possible when applying the method presented by Sas (2010).

7. Combining the LM with a flat word list

In many applications, except for a corpus that can be used to build stochastic n-gram LM, we have also the flat list

Table 4. Comparison of SWD error rates obtained with the original and modified language models.

	CT	TL	GM	PL	Total
n_w	2134	2370	2049	2417	8970
n_{DC}	307	238	213	265	1023
$n_d(LM)$	171	137	130	155	593
$n_f(LM)$	4	3	4	6	17
$e(LM)$	0.082	0.059	0.065	0.067	0.068
$\epsilon_e(LM)$	0.010	0.008	0.009	0.008	0.004
$n_d(LM_a)$	89	78	69	81	317
$n_f(LM_a)$	11	9	9	13	42
$e(LM_a)$	0.047	0.037	0.038	0.039	0.040
$\epsilon_e(LM_a)$	0.007	0.006	0.007	0.006	0.003
$\eta(LM_a)$	0.43	0.38	0.42	0.42	0.41
$n_d(LM_b)$	102	88	81	93	364
$n_f(LM_b)$	7	7	9	10	33
$e(LM_b)$	0.051	0.040	0.044	0.043	0.044
$\epsilon_e(LM_b)$	0.008	0.006	0.007	0.007	0.003
$\eta(LM_b)$	0.38	0.32	0.33	0.36	0.35
$n_d(LM_c)$	83	69	65	73	290
$n_f(LM_c)$	13	11	12	17	53
$e(LM_c)$	0.045	0.034	0.038	0.037	0.038
$\epsilon_e(LM_c)$	0.007	0.006	0.007	0.006	0.003
$\eta(LM_c)$	0.45	0.43	0.42	0.44	0.44

of *Out-Of-Corpus* (OOC) words that do not appear in the corpus, but belong to the general language glossary. Our aim is to include OOC words into the n-gram LM created in a typical way from the text corpus. The problem is how to combine the OOC word list with the LM, in particular how to estimate unigram probabilities for OOC words and n-gram probabilities for n-grams containing OOC words. Approximate unigram probabilities of OOC words can be acquired or they are not known at all. A typical case of an OOC word is the set of names (person names, surnames, city or street names, etc.) specified explicitly. In these cases, the approximate probabilities of OOC word occurrence can be derived from sources other than the corpus of texts, e.g., from databases containing records related to named entities. In other cases there is no explicit knowledge about OOC word occurrences.

We propose here to approximate their frequencies by applying a class-based approach to language modeling, which utilizes part-of-speech tagging. The concept of class-based modeling was primarily proposed by Brown *et al.* (1992) and then followed by other researches with various grouping criteria. It is not only unigram probabilities for OOC words that can be approximated in this way, but also their n-gram probabilities. One reasonable approach is to divide OOC and corpus words into classes according to their POS tags. Then n-gram probabilities for classes can be estimated in a typical way from the corpus and they can be applied to calculate word n-gram probabilities both for words appearing in the corpus and for OOC words.

Experiments presented by Niesler *et al.* (1998) show that a class-based approach utilizing POS tagging is not as efficient as application of categories based on stochastic properties of n-grams occurring in the corpus. In the case of the problem being considered here, we cannot apply the latter approach to OOC words because they are not present in the corpus. Therefore we based the solution merely on POS grouping.

Let us consider the set of POS classes $\{C_1, \ldots, C_K\}$ corresponding to various parts of speech and to specific inflectional forms (e.g., case, plural/singular forms for nouns). Let $\overline{c}(w)$ denote the set of POS classes for the word w. Due to the POS tagging ambiguity, this set may consist of more than one class. The set of POS classes can be determined from the corpus and OOC words automatically. For the precise context-dependent tagging of Polish words, we can use tools described by Piasecki (2007) as well as Piasecki and Radziszewski (2008). As another option, the simpler tool *Morfeusz* described by Woliński (2006) can be applied to find grammatical categories of isolated words. The class-based model LM_{CB} is then created using the standard method described by Niesler *et al.* (1998). The standard word n-gram model LM_W is also created using the same corpus. The model LM_W is then extended by OOC word inclusion in two steps. In the first one, the total probability of all OOC words occurrences is estimated and it is discounted from the probability mass assigned to unigrams actually occurring in the corpus. In the second step, the discounted probability mass is redistributed among OOC words.

In order to assign non-zero unigram probabilities to OOC words, the fraction of prior probabilities of words occurring in the corpus is discounted and the saved probability mass is distributed among OOC words. It is reasonable to approximate the discounted probability as the probability that a word in the utterance is an *Out-Of-Vocabulary* (OOV) word[3]. This probability can be estimated by a simple experiment where the text corpus utilized in the LM creation procedure is used again. Obviously, the stochastic properties of the corpus will not be changed significantly if we extract a single sentence from it. For words in the extracted sentence, checks are made if they appear in the remaining part of the corpus.

By applying this experiment to all individual sentences in a leave-one-out manner, we can count the number of missing word occurrences n_f. The overall probability p_f of OOV word occurrence can be then approximated as

$$p_f = \frac{n_f}{n_T + n_f}, \tag{35}$$

where n_f is the count of words that occur in the corpus only once and n_T is total number of word occurrences in the corpus. Let us assume that OOC words cover an arbitrary assumed fraction δ of all OOV words. Therefore, the probability mass that will be assigned to OOC words will be δp_f, and the same mass of probability must be discounted from unigram probabilities of the words in the corpus. The updated probabilities of corpus words $p'(w)$ can be calculated as $p'(w) = p(w)(1 - \delta p_f)$, where $p(w)$ is the prior word probability in the primary LM_W.

The discounted probability mass δp_f is distributed among POS classes resulting in class residual probabilities $p_r(c_i)$. The residual probabilities are proportional to the corresponding class probabilities computed in the LM_C model. Only these classes having their members in OOC word set are considered. Probabilities $p_r(c_i)$ are finally redistributed among OOC words belonging to them. OOC words are assigned to POS classes using their POS tagging. The words, however, often cannot be assigned to a unique POS class unambiguously. If a word is assigned to various POS classes in the corpus, then it should be given higher probability than the word assigned only to a single class. It leads to the following formula for the final OOC word

[3]OOV is not quite the same as OOC. By OOC we denote the words from a finite, explicitly given list, while OOV is the set of all words belonging to the language that do not occur in the corpus: $OOC \subseteq OOV$.

unigram probability $p_{OOC}(w)$:

$$p_{OOC}(w) = \delta p_f \frac{\sum_{c\in\overline{c}(w)} p(c)/n_c}{\sum_{c\in\Xi(OOC)} p(c)}, \tag{36}$$

where c is the symbol of the POS class, p_c is the POS class probability determined using LM_{CB}, $\overline{c}(w)$ denotes the set of POS classes to which the word w was assigned by the tagger, n_c is the number of OOC words tagged with the class c (i.e., $n_c = \mathrm{card}\{w : w \in OOC \wedge c \in \overline{c}(w)\}$) and $\Xi(OOC)$ is the set of POS classes that appear at least once as a tag of an OOC word. Because we have no information about OOC word frequencies other than that resulting from POS classification, we assume that each occurrence of a word in the POS class is equally probable.

In the resultant LM, bigrams (w_i, w_{i+1}), where $w_{i+1} \in OOC$, do not appear explicitly because such word pairs were by definition not encountered in the corpus. Therefore, similarly as in the case of other word pairs not occurring in the corpus, the bigram probability for word pairs containing an OOC word is calculated by backing off to unigram probability.

7.1. Empirical evaluation. The aim of the experiment described here is to compare the proposed method of OOC word list inclusion into the LM with a simpler approach, where all unigram probabilities of all new words are set equal each to other. In this rival method, the mass of discounted probability is determined also by estimating the probability of OOV word occurrence, but next it is distributed uniformly among OOC words.

The class set was created based on the POS assignment to words and their inflectional features specific for individual parts of speech. For nouns and adjectives, 42 classes were created based on the case, number and gender. 18 classes were created for verbs based on the tense, person and number. 35 classes were created for numerals. For the remaining parts of speech, a single class was used for each individual part. The *Morfeusz* program was used for assigning POS tags to words. *Morfeusz* is based on a database consisting of verified words tagging. Therefore, it is not able to assign tags to new words not registered in its database. In our experiment, for words not recognized by *Morfeusz*, the classes were determined based on other correctly tagged words having the same longest suffix. If $\sigma(w)$ is the set of words sharing the longest suffix with the word w, then the set of classes $\overline{c}(w)$ containing w is determined as

$$\overline{c}(w) = \bigcup_{v\in\sigma(w)} \overline{c}(v) \tag{37}$$

The data for the experiment were prepared in the following way. First, the fraction of the available corpus was excluded from the text set used later to create the baseline LM. This fraction was selected so as to contain all occurrences of the least frequently occurring words. The OOC word list was created from all words in the excluded set that do not occur in the remaining part of the corpus. The test set was created be selecting such utterances from the excluded set which contained at least one OOC word. The utterances selected in this way were then recorded as speech samples and used to test the recognizer performance. The baseline LM was prepared using the remaining part of the corpus. Finally, this model was extended with OOC words using two methods being compared (uniform OOC word probabilities and OOC probabilities computed using POS grouping). The experiment was carried out for 4 domains: CT, TL, GM and PL, presented in Section 5. For each domain, OOC words recognition accuracy was computed as

$$Acc(LM) = \frac{n_{OOC} - n_e(LM)}{n_{OOC}}, \tag{38}$$

where n_{OOC} is the count of OOC word occurrences in test utterances and $n_e(LM)$ is the count of OOC words recognized incorrectly using the LM.

Results for the LM created by uniform probability distribution among OOC words (LM_U) and for the LM created using the class-based approach (LM_{CB}) are shown in Table 5. The first row contains counts of words n in the test sets. The counts of OOC words occurrences n_{OOC} are given in the second row. The meaning of the symbols $Acc(LM)$ and $n_e(LM)$ is as in the formula (38). For comparison, the 3 bottom rows contain results obtained using the test set consisting entirely of words occurring in the text corpus and the recognizer based on the unigram language model (LM_{ug}). The confidence intervals of the estimated accuracies were determined at the confidence level $1 - \alpha = 0.9$ and are denoted by $\epsilon_{Acc}(LM)$.

It can be observed that application of the class-based approach in computing unigram probabilities of OOC words does not lead to a significant improvement of

Table 5. Speech recognition accuracy of utterances containing OOC words.

	CT	TL	GM	PL	Total
n	9540	11076	10767	12699	44082
n_{OCC}	1678	2270	1939	2151	8037
$n_e(LM_U)$	120	236	173	257	768
$Acc(LM_U)$	0.928	0.896	0.911	0.880	0.902
$\epsilon_{Acc}(LM_U)$	0.010	0.011	0.011	0.011	0.005
$n_e(LM_{CB})$	113	228	177	238	756
$Acc(LM_{CB})$	0.933	0.899	0.908	0.889	0.906
$\epsilon_{Acc}(LM_{CB})$	0.010	0.010	0.011	0.011	0.005
$n_e(LM_{ug})$	105	247	141	208	701
$Acc(LM_{ug})$	0.937	0.891	0.927	0.903	0.913
$\epsilon_{Acc}(LM_{ug})$	0.010	0.011	0.010	0.011	0.005

the OOC word recognition accuracy in comparison with application of the simpler method, where OOC words are given uniformly distributed probabilities. The confidence intervals determined for the estimated accuracies of the compared models overlap strongly. This indicates that the models obtained using the compared methods are not significantly different. The achieved OOC recognition accuracy is very close to the overall recognition accuracy obtained with a completely flat unigram language model. This observation is consistent with the conclusion drawn in Section 5 that in Polish (and probably in other loose word order languages) strongly smoothed language models exhibit good properties in ASR.

8. Boosting the probability of very important phrases

In some cases of ASR applications, there are especially significant or frequently used *Very Important Utterances* (VIU) that should be recognized with very high accuracy (e.g., commands interleaved with text being dictated to the ASR system). Recognition errors in such utterances are particularly annoying and users insist on improving recognition accuracy of these utterances, even at the expense of slight reduction of the accuracy of other utterances. The obvious way to achieve this goal is to (a) add artificially many instances of important utterances to the text corpus used to build the LM or (b) increase significantly probabilities of n-grams constituting the utterance. Radical modification of the LM aimed at selected utterances may, however, lead to a significant degradation of recognition of other utterances consisting of words phonetically similar to those appearing in important utterances. The problem is therefore how to modify the LM so as to assure the probability of correct VIU recognition at least at the specified level α of accuracy, while minimizing the degradation of other utterances' recognition quality.

Let us define the problem more formally as follows. Let $U = \{u_1, \ldots, u_m\}$ denote the set of VIUs where each VIU is a sequence of words $u_i = (w_{i_1}, w_{i_2}, \ldots, w_{i_{l_i}})$. In particular, the sequence may contain just a single word. We assume that the LM used in speech recognition contains all words appearing in the set U. Our aim is to keep VIU-related recognition error rate at the specified level α. The VIU-related error consists in erroneous recognition of VIU utterances and in recognizing other utterance as one of VIUs. The first type error probability for the single utterance $u \in U$ is

$$p_{e_I}(u) = P(\Psi(O(v); LM)) \neq u | v = u), \tag{39}$$

while the second type error probability is

$$p_{e_{II}}(u) = P(\Psi(O(v); LM)) = u | v \neq u) = \frac{\sum_{v \in W^+} p(v) P(\Psi(O(v); LM)) = u)}{\sum_{v \in W^+} p(v)}. \tag{40}$$

$\Psi(O(v); LM)$ denotes here the recognizer applied to the observation sequence $O(v)$ extracted from the acoustic signal of the utterance v. The LM is the language model used by the recognizer. W^+ denotes the set of all word sequences consisting of words appearing in the LM, excluding words being VIUs. In the approach presented here, the suppression of the VIU-related error rate is achieved by modifying the LM so that, for each $u \in U, p_{e_I}(u) < \alpha$ and $p_{e_{II}}(u) < \alpha$. The error consisting in recognition of one VIU as another VIU is neglected. This is motivated by the fact that in a majority of applications VIUs can be selected intentionally so as to minimize mutual acoustic similarity among them. The method of VIU set selection that minimizes VIUs mutual similarity is presented by Sas (2009).

The method of second type error probability computation defined in Eqn. (40) is formally well-founded but troublesome in practice due to the summation over all possible word sequences W^+. In practice it can be simplified by summing only over the relatively small set $q(u)$ of utterances that are likely to appear in real speech or are likely to be misrecognized as u. Thus, instead of considering the overall second type error probability as defined in (40), only the error probability conditioned on the set $q(u)$ can be considered:

$$\widehat{p}_{e_{II}}(u) = \frac{\sum_{v \in q(u)} p(v) P(\Psi(O(v); LM)) = u)}{\sum_{v \in q(u)} p(v)}. \tag{41}$$

The VIU can be either a single word or a sequence of words. In the latter case, the VIU can be replaced by a corresponding multi-word, which will be added to the vocabulary of words in the model. Hence, hereafter we will assume that VIUs u_i are single words. In most applications it seems to be reasonable to assume that VIUs are uttered as isolated utterances. Thus, a method of LM modification can be used that is similar to that one proposed in the previous section, where some discounted unigram probability was redistributed among multi-words. Now, however, according to the assumed isolation of VIUs, more appropriate approach is to redistribute bigram probabilities $p(w|\langle s \rangle)$ for bigrams $(\langle s \rangle, w)$ containing words w that appear in the initial LM. The symbol $\langle s \rangle$ denotes here the pseudo-word representing the beginning of the utterance. The discounted probability will be assigned to bigrams $(\langle s \rangle, u)$ for $u \in U$.

The unigram probabilities for words occurring in the corpus do not need to be modified. If VIUs are new words or multi-words, then for formal LM consistency they must

be added to unigrams. However, their probabilities are set to zero. This is because we assume here that VIUs will be always recognized as isolated utterances, so they can occur only as successors of the pseudo-word $\langle s\rangle$. In the case of a bigram LM, the unigram probabilities are necessary only in order to compute the probability if the bigram is not explicitly contained in the LM, i.e., if the successor word w_{i+1} does not belong to the set $\mathcal{N}(w_i)$ specified in Section 4. In such a case, the backoff to unigrams is applied according to the formula (25). The explicit bigrams $(\langle s\rangle, u)$ for all $u \in U$ will be added to the LM. For all other preceding words, the appearance of VIU as a successor is forbidden, which will be achieved by setting VIU unigram probabilities to zero.

We will need to discount the probabilities $p(w|\langle s\rangle)$ for bigrams represented explicitly in the initial LM and to redistribute the discounted probability mass among bigrams containing VIUs as successors. In order to preserve the model properties related to other utterances, our aim is to modify the model as little as possible so as to achieve the assumed VIU recognition accuracies. Thus the problems to be solved are the following:

(a) how much of the probability mass should be discounted from the probabilities $p(w|\langle s\rangle)$;

(b) how the discounted probability mass should be redistributed among the probabilities $p(u|\langle s\rangle), u \in U$.

Alternatively, the problem is: What should the minimal probabilities $p(u|\langle s\rangle)$ be which yield VIU-related error within the assumed interval $(0, \alpha)$.

A theoretical solution to the optimal selection of VIU bigram probabilities is difficult due to the complexity of the computations involved in the Viterbi procedure (Young and Everman, 2009; Lee *et al.*, 2001) typically applied in HMM-based speech recognizers. A purely empirical approach is not feasible in most cases either due to the necessity to collect a big number of VIU acoustic samples. Therefore we propose here a procedure that utilizes the existing acoustic model as a kind of artificial speech sample synthesizer.

Let $r(u)$ denote the set of speech samples created by uttering the VIU u, and let R denote the sum of sets $R = \bigcup_{u\in U} r(u)$. Let us denote by $g(u)$ the set of speech samples obtained by uttering other utterances out of U that are likely to be incorrectly recognized as u. The sets $r(u)$ and $g(u)$ constitute the verification set used in order to modify the LM appropriately. The VIU-related errors $p_{e_I}(u)$ and $\widehat{p}_{e_{II}}(u)$ can be estimated empirically as

$$\overline{p}_{e_I}(u) = \frac{n_{e_I}(u)}{\mathrm{card}(r(u))}, \qquad \overline{p}_{e_{II}}(u) = \frac{n_{e_{II}}(u)}{\mathrm{card}(g(u))}, \tag{42}$$

where $n_{e_I}(u)$ is the count of samples from the set $r(u)$ incorrectly recognized, and $n_{e_{II}}(u)$ is the count of samples from the set $g(u)$ incorrectly recognized as u. VIU-related error approximation can be used in the iterative procedure of LM modification. In the i-th iteration it finds the minimal bigram probability $p(u_i|\langle s\rangle)$ that leads to the error estimate on the test set within the assumed interval.

Probabilities established once for already processed VIUs are not changed in the later steps of the procedure, so the necessary discounted probability mass is always obtained from other bigrams representing words actually appearing in the corpus and not being VIUs. Application of the probability estimates (42) instead of the true probabilities (39) and (41) is acceptable when they are precise. Of course, this requires many speech samples in sets $r(u)$ and $q(u)$. In the method proposed here the real utterances were replaced by observation sequences produced by the HMM model used as a random automaton that produces observation sequences.

The utterances represented in the sample set $g(u)$ should be drawn from the corpus, so that they are likely to appear in the real utterances. To make the second type error estimation feasible, the amount of testing utterances must be kept within the reasonable limit. For this reason, the set $g(u)$ should consist of utterances that are likely to be misrecognized when $O(u)$ is presented to the speech recognizer. Most of speech recognizers provide the option to deliver not only the most likely words sequence but rather a list of N-best candidates. In the method presented here, the word sequences used to create the set $g(u) \subset W^+$ are obtained by gathering N-best word sequences not being the actually spoken VIU obtained when recognizing speech samples from the set $r(u)$. In this way, the set of word sequences $q(u)$ is obtained. Then for each sequence from $q(u)$ the corresponding random HMM automaton is configured. By running this automaton randomly, an unlimited number of artificial speech samples constituting the set $g(u)$ can be created. By Q we will denote the set of verification utterances created in this way.

The misrecognition of an utterance is the result of its acoustic similarity to other sequences of words. The likelihood of VIU misrecognition is used to order them in the procedure of LM modification. Misrecognition likelihood can be evaluated by acoustic similarity to the most similar word sequence that can be constructed from the words in the LM vocabulary. The most likely misrecognition results can be taken from the set $q(u)$ obtained as described above. For each $u \in U$ its similarity to all elements of $q(u)$ can be evaluated, and finally the misrecognition likelihood can be computed based on the similarity to the most similar misrecognized sequence of words. For the sake of this method, acoustic similarity of two utterances is calculated as the edit distance (Levenshtein distance) between sequences of phones obtained as phonetic transcriptions of the utterances. The original edit distance, being the number of

insertion, deletion and substitution operations, is modified so as the substitutions of various types of phones are assigned various weights corresponding to their acoustic similarities.

The complete procedure applied here to modify the input LM can be defined as Algorithm 1. It iteratively increases the bigram probability $p(u|\langle s\rangle)$ for the utterance $u \in U$ being currently processed until the first type error probability falls within the required interval or the second type error probability goes out of it. The permissible bigram probability interval is $(p_{\min}, p_{\max})$, where $p_{\min} = 1/k$, $p_{\max} = 1/2m)$, k is the number of words in the vocabulary and m is the number of VUIs in the set U.

Algorithm 1. LM modification for correct VIU recognition.

create the set of observation sequences R and Q
  using the random automaton based on LM/AM;
order the set U by VIU importance;
for all $u \in U$ in decreasing order **do**
  for $p = p_{\min}; p \leq p_{\max}$; step δ **do**
    LM' = LM;
    set $p(u|\langle s\rangle) = p$ in LM';
    discount the probability p from
      bigrams $p(w|\langle s\rangle), w \notin U$;
    calculate estimates $\overline{p}_{e_I}(u), \overline{p}_{e_{II}}(u)$ using LM';
    if $\overline{p}_{e_I}(u) < \alpha$ or $\overline{p}_{e_{II}}(u) > \alpha$ **then**
      break;
    end if
  end for
  $LM = LM'$;
end for
return LM;

8.1. Empirical evaluation. The proposed method is based on the assumption that the accuracy of recognition of the artificially created utterances is close to the accuracy achieved for the human speaker. The aim of the experiment is to verify this assumption. We also want to test how the boosting of VIU bigram probability impacts the recognition of other acoustically similar utterances. In the experiment, the speaker-independent, gender specific acoustic model was used. We tested the set of VIUs being commands that control the editing of the text being dictated. The set of commands in Polish and their translations into English are listed in Table 6. The test set was created by uttering commands by the set of speakers. Each command was uttered 20 times by each of the 6 speakers (3 female, 3 male).

Then the initial LM was modified using the tuning procedure described in the previous section to achieve the VIU recognition error rate at the level of $\alpha \in (0.01, 0.07)$. The actual accuracy of VIU recognition was then evaluated using the set of test utterances. The results are shown in Fig. 1.

It can be observed that the actual error rate is greater than the expected one based on testing with artificial utterances. This can be explained by the fact that the acoustic model does not simulate the speech perfectly, so the error rate for real speech is higher than that evaluated using the model which is also used in the recognition procedure. The dependence of the predicted and actual error rate is, however, almost linear, apart from the interval $(0.1, 0.2)$, where the assumed error rate cannot be achieved due to an increasing second type error. The experiment shows that in order to obtain the assumed first type error close to α' the model should be tuned for $\alpha \approx 0.7\alpha'$.

Decreasing first type error $p_{eI}(u)$ by increasing the probability $p(u|\langle s\rangle)$ leads to the increase of the second type error consisting in erroneous recognition of other utterances as u. The proposed algorithm prevents excessive increase of the second type error by limiting the bigram probability $p(u|\langle s\rangle)$ if $p_{eII}(u)$ exceeds assumed

Table 6. Set of VIUs used in the experiment.

Command utterance	Translation into English
Wyczyść	Clear
Cofnij	Back
Zakończ	Terminate
Zapisz	Save
Nowa linia	Insert the new line
Zapisz do pliku	Save to a file
Zaznacz słowo	Select word
Na koniec	Go to the end
Na początek	Go to the beginning
Następne słowo	Go to the next word
Poprzednie słowo	Go to the previous word
Duże litery	Upper case letters
Małe litery	Lower case letters

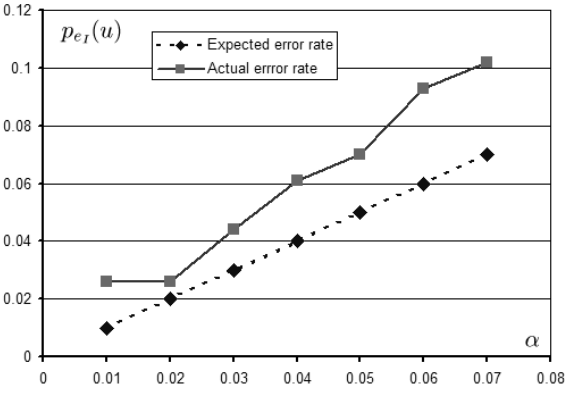

Fig. 1. First type actual error $\overline{p}_{eI}(u)$ vs. the expected error level α.

limit α. In Fig. 1 this can be observed for $\alpha < 0.02$, where the actual first type error is not further reduced because the second type error exceeds its limit. When LM is being updated in the iterative algorithm, it uses the estimates of first and second type errors based on the artificial utterances. The legitimacy of artificial utterance application when estimating the second type error was verified experimentally. In the experiment, second type error estimation based on artificial utterances was compared with its estimate based on actual human-spoken utterances from $q(u)$ sets. Estimates obtained with modified LMs created for various values of $\alpha \in (0.01, 0.07)$ were compared. Results are shown in Fig. 2.

The results obtained in experiments related to the VIU are obviously specific for the VIU set and the LM being used. For other VIUs and/or LMs, the relation of the actual accuracy to that obtained with artificial test sets as well as the relation of the first type to the second type errors can be different. We believe, however, that general tendencies demonstrated in this particular case are also valid in different conditions. Thus achieving the sufficiently low VIU related error using the proposed method is possible without deteriorating ASR accuracy for other utterances.

9. Conclusions and future work

In this paper, we focused on the problems of constructing language models for automatic speech recognition in Polish. The main aim of the described works was to take into account the specific features of the Polish language when building the language models, such as rich inflection, loose word order, or frequent appearance of short words that exhibit the tendency to be falsely removed in the recognized sentence. We also took into account practical issues often encountered when constructing the language model for the specific application: the need to extend the model with words

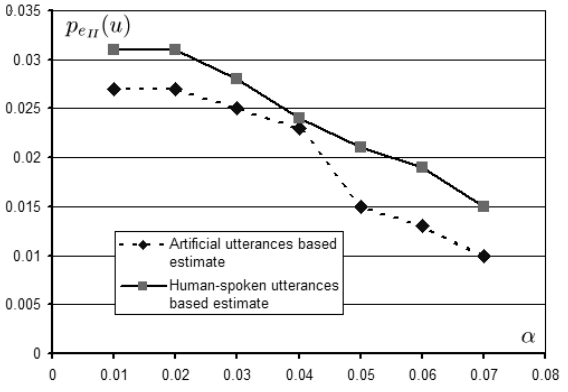

Fig. 2. Comparison of second type error estimates obtained with artificial and human-spoken utterances.

that did not appear in the text corpus and the need to recognize the small set of selected utterances with very high accuracy. We proposed the pipelined model construction method, where the initial model is created at the first stage and then extended or improved at subsequent stages.

At the first stage we tested various smoothing methods used when building the initial language model: absolute discounting, Good–Turing, Kneser–Ney and modified Kneser–Ney smoothing. The tests were carried out for four subdomains of Polish speech that differ in complexity (available amount of texts in the corpus, size of the dictionary). The tests showed that there are no significant differences in speech recognition accuracy obtained when applying language models constructed using the compared techniques. The Kneser-Ney method showed marginally better performance than other methods, hence it was used as the baseline at further stages of model building.

At the next stage, the method that takes into account the loose word order in Polish was applied to modify the model. The method boosts the probabilities of bigrams which were observed at distant positions in sentences belonging to the corpus. As a result, we obtained small but observable improvement in recognition accuracy in domains where model perplexity is high and speech recognition is difficult. In the case of simple models, where perplexity is low, model modification resulted in worsening recognition accuracy. Hence, the practical conclusion is that the bigram probability boosting of distant co-occurring words should be applied only to models of high perplexity.

The model modification introduced at the third stage is aimed at avoidance of short word deletion. Our experiences with practical application of ASR to Polish speech showed that it is one of the most common errors. The methods proposed here achieve the goal by modifying the language model only. They were compared with another method (proposed earlier by the same author) which carries out short word deletion correction as the postprocessing stage. The performance of all compared methods is similar. Short word deletion error rate relative reduction is at the level of 40%. The methods proposed here are, however, easier in application, because the effect can be obtained by merely modifying the language model. As a result, the method can be used in any ASR system, without the need to modify the system logic.

The aim of the next stage is to extend the model with additional words not appearing in the corpus. Two methods of model extension with the flat list of additional words were compared. The first, simpler method assigns equal unigram probabilities to all new words. The alternative method assigns probabilities to new unigrams using part-of-speech tagging. Experimental evaluation showed that there are no statistically significant

differences between the compared methods. Hence the practical recommendation is to use the method that assigns uniform probabilities to new words, due to its simplicity.

At the last stage a method was proposed that modifies the model so as to assure that the selected utterances are recognized with an arbitrarily high level of accuracy. The experiments carried out with the set of important utterances being the spoken text editor commands proved that the goal can be achieved without significant degradation of other utterances recognition accuracy.

The overall conclusion following from the experiments described here is that the Polish language is hard for speech recognition due to its specific features. Application of methods focused specifically at individual sources of difficulties may improve recognition accuracy. The procedure of language model construction presented here is an example of such an approach. Although the experiments were carried out for the Polish language, we believe that similar results can be obtained for other languages of similar properties, e.g., for Slavonic ones.

The concept of the language construction pipeline can be extended with more stages aimed at other language features or specific type of errors in speech recognition. A promising direction seems to be at application of class-based models combined with ordinary word-based ones. A class-based approach relying on POS tagging can be used to exclude specific sequences of words that are almost impossible to appear in spoken language. This can be useful in eliminating the error consisting in successive appearance of short conjunctions or prepositions. This type of error often occurs in utterance fragments being short pauses not detected at the acoustic level.

Experiments in pattern recognition prove that application of classifier combinations can increase recognition accuracy (Woźniak and Krawczyk, 2012). The classifier combination concept can be also applied to ASR. One of the possibilities is to combine classifiers based on LMs created using various corpora or construction techniques. Model combination can be also used at the level of LM construction for a single classifier. A direction that seems to be insufficiently explored in the case of Polish ASR, is a model construction by combining simpler models. It can be applied to the construction of a domain specific language model by combining a model created from a relatively small corpus of domain-specific texts with a domain-independent model built using a bigger corpus.

References

Brown, P., deSouza, P.V., Mercer, R.L., Pietra, V.J.D. and Lai, J.C. (1992). Class-based n-gram models of natural language, *Computational Linguistics* **18**(1): 467–479.

Brychcin, T. and Konopik, M. (2011). Morphological based language models for inflectional languages, *Proceedings of the 6th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems, Praque, Czech Republic*, pp. 560–563.

Chen, S. and Goodman, S. (1999). An empirical study of smoothing techniques for language modeling, *Computer Speech and Language* **1**(13): 359–394.

Chen, Y. and Chan, K. (2003). Extended multi-word trigger pair language model using data mining technique, *Systems, Man and Cybernetics* **1**(1): 262–267.

Devine, E., Gaehde, S. and Curtis, A. (2007). Comparative evaluation of three continuous speech recognition software packages in the generation of medical reports, *Journal of American Medical Informatics Association* **1**(7): 462–468.

Gale, A. and Sampson, G. (1995). Good–Turing frequency estimation without tears, *Journal of Quantitative Linguistics* **2**(1): 217–239.

Goodman, J. (2001). A bit of progress in language modeling extended version, *Technical Report MSR-TR-2001-72*, Machine Learning and Applied Statistics Group, Microsoft Research, Redmond, WA.

Iyer, R. and Ostendorf, M. (1999). Modeling long distance dependence in language: Topic mixtures versus dynamic cache models, *IEEE Transactions on Speech and Audio Processing* **7**(1): 30–39.

Jelinek, F., Merialdo, B., Roukos, S. and Strauss, M. (2001). A dynamic language model for speech recognition, *Proceedings of the Workshop on Speech and Natural Language, HLT'91, Pacific Grove, CA, USA,* pp. 293–295.

Jurafsky, D. and Matrin, J. (2009). *Speech and Language Processing. An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition*, Pearson Prentice Hall, Englewood Cliffs, NJ.

Kasprzak, W., Wilkowski, A. and Czapnik, K. (2012). Hand gesture recognition based on free-form contours and probabilistic inference, *International Journal of Applied Mathematics and Computer Science* **22**(2): 437–448, DOI: 10.2478/v10006-012-0033-6.

Katz, S. (1987). Estimation of probabilities from sparse data for the language model component of a speech recognizer, *IEEE Transactions on Acoustics, Speech, and Signal Processing* **35**(3): 400–401.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation, *MIT Summit 2005, Phuket, Thailand,* pp. 79–86.

Kolorenc, J., Nouza, J. and Cerva, P. (2006). Multi-words in the Czech TV and radio news transcription system, *Proceedings of SPECOM 2006, St. Petersburg, Russia*, pp. 70–74.

Lee, A., Kawahara, T. and Shikano, K. (2001). Julius—an open source real-time large vocabulary recognition engine, *Proceedings of the European Conference on Speech Communication and Technology (EUROSPEECH), Aalborg, Denmark*, pp. 1691–1694.

Mauces, M., Rotownik, T. and Zemljak, M. (2003). Modelling highly inflected Slovenian language, *International Journal of Speech Technology* **1**(6): 254–257.

Mikolov, T., Deoras, A., Kombrink, S., Burget, L. and Cernocky, J. (2011). Empirical evaluation and combination of advanced language modeling techniques, *INTERSPEECH, ISCA, Florence, Italy*, pp. 605–608.

Niesler, T., Whittaker, E.W.D. and Woodland, P. (1998). Comparison of part-of-speech and automatically derived category-based language models for speech recognition, *Proceedings of ICASSP 98, Seattle, WA, USA*, pp. 177–180.

Piasecki, M. (2007). Polish tagger TaKIPI: Rule based construction and optimisation, *Task Quarterly* **11**(1): 151–167.

Piasecki, M. and Broda, B. (2007). Correction of medical handwriting OCR based on semantic similarity, *in* H. Yin, P. Tino, E. Corchado, W. Byrne and X. Yao (Eds.), *Intelligent Data Engineering and Automated Learning—IDEAL 2007*, Lecture Notes in Computer Science, Vol. 4881, Springer-Verlag, Heidelberg, pp. 437–446.

Piasecki, M. and Radziszewski, A. (2008). Morphological prediction for Polish by a statistical *a tergo* index, *Systems Science* **34**(4): 7–17.

Sarukkai, R. and Ballard, D. (1996). Word set probability boosting for improved spontaneous dialogue recognition. The ab and tab algorithms, *Technical Report TR-601*, University of Rochester, New York, NY.

Sas, J. (2009). Optimal spoken dialog control in hands-free medical information systems, *Journal of Medical Informatics and Technologies* **13**: 113–120.

Sas, J. (2010). Application of local bidirectional language model to error correction in Polish medical speech recognition, *Journal of Medical Informatics and Technologies* **15**(1): 127–134.

Sas, J. and Żołnierek, A. (2011). Distant co-occurrence language model for ASR in loose word order languages, *Proceedings of the International Conference on Computer Recognition Systems Cores 2011, Wrocław, Poland*, pp. 767–778.

Vaiciunas, A., Kaminskas, V. and Raskinis, G. (2004). Statistical language models of Lithuanian based on word clustering and morphological decomposition, *Informatica* **15**(4): 565–580.

Ward, W. and Issar, S. (1996). A class based language model for speech recognition, *Acoustics, Speech, and Signal Processing, ICASSP 96, Atlanta, GA, USA*, pp. 416–418.

Whittaker, E. and Woodland, P. (2003). Language modelling for Russian and English using words and classes, *Computer Speech and Language* **17**(1): 87–104.

Woliński, M. (2006). Morfeusz—a practical tool for the morphological analysis of Polish, *Inteligent Processing and Web Mining: IIPWM 06, Ustroń, Poland*, pp. 503–512.

Woźniak, M. and Krawczyk, B. (2012). Combined classifier based on feature space partitioning, *International Journal of Applied Mathematics and Computer Science* **22**(4): 855–866, DOI: 10.2478/v10006-012-0063-0.

Young, S. and Everman, G. (2009). *The HTK Book (for HTK Version 3.4)*, Cambridge University, Cambridge.

Ziółko, B., Skurzok, D. and Ziółko, M. (2010). Word n-grams for Polish, *Proceedings of the 10th IASTED International Conference on Artificial Intelligence and Applications (AIA 2010), Innsbruck, Austria*, pp. 197–201.

Ziółko, J., Gałka, J., Jadczyk, T., Skurzok, D. and Masior, M. (2011). Automatic speech recognition system dedicated for Polish, *Proceedings of the INTERSPEECH 2011 Conference, Florence, Italy*, pp. 3315–3316.

Ziółko, J., Gałka, J. and Skurzok, D. (2010). Speech modelling using phoneme segmentation and modified weighted Levenshtein distance, *Proceedings of the ICALP2010 Colloquium, Bordeaux, France*, pp. 743–746.

Jerzy Sas received his Ph.D. degree from the Institute of Biocybernetics and Biomedical Engineering of the Polish Academy of Sciences in 1993. Currently he is an assistant professor at the Institute of Informatics, Wrocław University of Technology. His research focuses on speech recognition, medical informatics and photorealistic computer graphics. Doctor Sas has published over 90 papers. He has been involved in many research and developments projects related to medical informatics. He was also the leader of the development team of a commercially available speech recognition system for Polish aimed at medical and mobile applications.

Andrzej Żołnierek works at the Department of Systems and Computer Networks, Faculty of Electronics, Wrocław University of Technology. He received his Ph.D. degree from the Institute of Biocybernetics and Biomedical Engineering of the Polish Academy of Sciences in 1986. His main interests are in pattern recognition and artificial intelligence and their applications, as well as in image and speech analysis in medical diagnosis support. He has published over 40 papers. He has been involved in projects related to application of pattern recognition to medical diagnosis.

Received: 10 March 2012
Revised: 18 March 2013