INTERNATIONAL JOURNAL OF SPEECH TECHNOLOGY 8(4), 341–361, 2005

DOI: 10.1007/s10772-006-9047-5

Scaling Smoothed Language Models

A. VARONA AND I. TORRES

Dpto. Electricidad y Electrónica, Fac. Ciencia y Tecnologia, Basque Country University, Barrio Sarriena s/n.
48940 Leioa, Vizcaya (Spain)
amparo@we.lc.ehu.es
manes@we.lc.ehu.es

Abstract. In Continuous Speech Recognition (CSR) systems a Language Model (LM) is required to represent the syntactic constraints of the language. Then a smoothing technique needs to be applied to avoid null LM probabilities. Each smoothing technique leads to a different LM probability distribution. Test set perplexity is usually used to evaluate smoothing techniques but the relationship with acoustic models is not taken into account. In fact, it is well-known that to obtain optimum CSR performances a scaling exponential parameter must be applied over LMs in the Bayes' rule. This scaling factor implies a new redistribution of smoothed LM probabilities. The shape of the final probability distribution is due to both the smoothing technique used when designing the language model and the scaling factor required to get the optimum system performance when integrating the LM into the CSR system. The main object of this work is to study the relationship between the two factors, which result in dependent effects. Experimental evaluation is carried out over two Spanish speech application tasks. Classical smoothing techniques representing very different degrees of smoothing are compared. A new proposal, Delimited discounting, is also considered. The results of the experiments showed a strong dependence between the amount of smoothing given by the smoothing technique and the way that the LM probabilities need to be scaled to get the best system performance, which is perplexity independent in many cases. This relationship is not independent of the task and available training data.

Keywords: continuous speech recognition, language models, smoothing techniques, scaling factors

1. Introduction

Continuous Speech Recognition (CSR) systems require a Language Model (LM) to integrate the syntactic and/or semantic constraints of the language. The goal of an LM is to estimate the *a priori* probability $P(\Omega)$ of a sequence of words $\Omega \equiv \omega_1\omega_2\ldots\omega_{|\Omega|}$ to be pronounced. The most classical statistical methods for generating LM's are based on the estimation of the probability of observing a word given the $n-1$ preceding lexical units (n-gram models): $P(\omega_i|\omega_1\ldots\omega_{n-1})$ (Rosenfeld, 2000). However, there are a high number of sequences of words that do not appear in training corpora (unseen events) and could appear in tests. Thus, a certain mass of probability must be subtracted from the seen combinations and redistributed among the unseen ones, i.e, a smoothing technique must be applied (Ney et al., 1997; Chen and Goodman, 1999; Chen and Rosenfeld, 2000).

The test set perplexity (PP) is typically used to evaluate the quality of the LM (Ney et al., 1997; Chen and Goodman, 1999) and the quality of the smoothing technique. Perplexity can be interpreted as the (geometric) average branching factor of the language according to the model. It is a function of both the task and the model. It is supposed that the "best" models get the "lowest" Word Error Rates (WER) in the CSR system, but there are many contra examples in literature (Rosenfeld, 2000). The ability of the test set perplexity to predict the real behavior of a smoothing technique when the smoothed LM is working into a CSR system could be questioned (Clarkson and Robinson, 1999) since it does not take into account the relationship with acoustic models. Several attempts have been made to devise metrics that are better correlated with Word Error Rates than perplexity (Clarkson and Robinson, 1999; Bimbot et al., 2001), but for now perplexity remains the main metric for practical language model construction (Rosenfeld, 2000). In fact, the quality of the model must be ultimately measured by its effect on the specific task for which it was designed, namely by its effect on the system error rate. However, error rates are typically non-linear and poorly understood functions of language models (Rosenfeld, 2000). On the other hand, a recent work (Klakow and Peters, 2002) has shown good correlations between PP and WER when the task and available training data allow LM distributions close to the "true" distributions. In this paper we try to analyse the effect of the smoothing technique applied to the LM in the CSR system and to show its real impact on final system error rates.

CSR systems are invariably based on the well-known Bayes' rule, i.e., the recognizer must find the word sequence $\hat{\Omega}$ that satisfies (Jelinek, 1985):

$$\hat{\Omega} = \arg\max_{\Omega} P(\Omega) P(A \mid \Omega) \tag{1}$$

where $P(\Omega)$ is the *a priori* probability of the word sequence and $P(A \mid \Omega)$ is the probability of the sequence of acoustic observations given the sequence of words Ω. $P(A \mid \Omega)$ represent the acoustic likelihoods obtained through acoustic models, typically Hidden Markov Models (HMM), whereas $P(\Omega)$ are estimated by the LM.

However, it is well known that the best performance of a CSR system is obtained when LM probabilities are modified by introducing an exponential scaling factor (Jelinek, 1996; Rubio et al., 1997; Ogawa et al., 1998; Mangu and Stolcke, 2000). This parameter is needed because acoustic and LM probabilities are not real, but rather approximations, and are estimated from different knowledge sources (Jelinek, 1996). Other authors have stated that the language model weight compensates for the frame-independence assumption in HMM-based acoustic models (Mangu and Stolcke, 2000). In practice, the effect of the scaling factor is to attenuate all the LM probabilities in an exponential way. A new redistribution of the already smoothed LM probabilities is then achieved at decoding time. Thus, the shape of the final LM probability distribution depends on both the distribution previously provided by the smoothing technique and the score scaling applied at decoding time. Thus, the final optimum value of the scaling factor is not independent of the smoothing technique. Moreover, the effect of the smoothing technique on system performance is not independent of the subsequent scaling. The aim of this work is precisely to analyse the relationship between the two effects and to establish their related contribution to the final system performance. In Section 2 we fully explain this relationship and the motives behind our work.

Section 3 presents a brief summary of the *k*-Testable in the Strict Sense (*k*-TSS) languages (García and Vidal, 1990) used to generate LMs (Varona and Torres, 1999). *k*-TSS languages are a subclass of regular languages (García and Vidal, 1990). They can be considered as a syntactic approach of classical *n*-gram models derived from formal language theory.

Section 4 describes the syntactic back-off (Varona and Torres, 1999; Torres and Varona, 2001) smoothing techniques. The back-off formalism has been chosen in this work because the recursive scheme involved has been well integrated into syntactic formalism (Torres and Varona, 2001). Furthermore, the difference between recursively backing off to lower order *n*-grams (Katz, 1987) and linearly interpolating *n*-grams of different order (Jelinek and Mercer, 1980) matters only when we go into the details of the mathematical models (Ney et al., 1997). In this Section classical discounting-distribution schemes (Kneser and Ney, 1995; Clarkson and Rosenfeld, 1997; Ney et al., 1997; Chen and Goodman, 1999; Chen and Rosenfeld, 2000; Goodman, 2001) are defined under syntactic formalism. A new discounting method, Delimited discounting, is also presented. Delimited discounting keeps the well-known Katz's schema (Katz, 1987) but avoids the problem associated with the lack of an adequate distribution of the samples (Varona and Torres, 2000). The smoothing techniques described in this Section achieve very different probability distributions, resulting in a wide range of smoothed models, from very low to very high smoothed LMs. This point, which concerns the main goal of our work, is also analyzed in Section 4 when we describe each discounting-distribution schema.

Section 5 presents the results of the experiment in terms of both classical test set perplexity and CSR system performance. The WER obtained through the experiments as well as the computational cost involved are considered in evaluating the CSR system performance. Experiments were carried out over two Spanish databases of very different difficulty. The first one, called Bdgeo (Díaz et al., 1998), is a task oriented speech corpus consisting of a medium size vocabulary. It was specially designed to test speech understanding systems with medium difficulty application tasks. The second, called Info-tren (Bonafonte et al., 2000; Rodríguez et al., 2001a), was collected from real users of a human-machine dialogue system. It includes spontaneous speech and, thus, results in a high difficulty application task.

Finally, some concluding remarks are given in Section 6.

2. Introducing the LM into the CSR System

Within a CSR system there are several heuristic parameters that must be adjusted to obtain optimum performances, such as the beam-search factor to reduce the computational cost, etc. But, the most important factor to be optimized, due to its great effect on final CSR system performance (Jelinek, 1996; Rubio et al., 1997; Ogawa et al., 1998; Mangu and Stolcke, 2000; Varona and Torres, 2001, 2003), is the scaling factor α applied over LM probabilities $(P(\omega))^{\alpha}$. From a theorical point of view, this parameter is needed because acoustic and LM probability distributions are not real but approximations (Jelinek, 1996). The two probability distributions in Baye's rule (Eq. (1)) are estimated independently using different stochastic models that represent different knowledge sources. Moreover, the parameters of the acoustic and language models are estimated on the basis of speech and text data corpora, respectively. Each corpus was designated with a different purpose, and therefore has a different vocabulary, size, complexity, etc. Thus, a balance parameter α needs to be applied to reduce these effects and then obtain good system performance. Other authors have stated that the language model weight compensates for the frame-independence assumption in HMM-based acoustic models, which underestimate the joint likelihoods of correlated acoustic observations (Mangu and Stolcke, 2000).

In practice, acoustic and LM have very different ranges of values. The accumulated probabilities at the end of each partial hypothesis in the Viterbi trellis is a combination of acoustic and language model probabilities. Acoustic probabilities are usually smaller than language probabilities and are applied many more times. The gap among accumulated probabilities is therefore usually bigger than the gap among LM probabilities. The immediate consequence is that LM probabilities are irrelevant in most situations for choosing the best path[1] (Varona and Torres, 2001). However, LM probabilities are attenuated when they are raised to a power $\alpha > 1$, but this attenuation is higher for lower probability values. A bigger gap is therefore obtained between high and low probabilities and the LM probabilities are then more relevant for deciding the next word combination.

At this point it is important to notice that the above LM probability distribution depends on several factors such as the language, the task, the training corpus composition, the language model order, etc. But the smoothing defines a redistribution of the LM probabilities by subtracting a certain mass of probability from the *seen events* and by redistributing it among the *unseen* ones according to the specific technique and discounting schema. As a consequence, the final optimum value of the α scaling factor cannot be independent of the smoothing technique. Moreover, the effect of the smoothing technique on the system performance is not independent of the subsequent scaling achieved at decoding time. Thus, it cannot be measured in terms of test set perplexity. The relationship between the LM probability redistribution achieved by the smoothing technique discounting schema and the probability redistribution achieved at decoding time when weighting the LM probabilities to be established. Our main motivation is to analyse the relationship between the two effects and to establish their related contribution to the final system performance.

It is well known that smoothing is a central issue in language modeling, and thus, in CSR system construction and performance. However, all the comparisons between smoothing techniques, even the most thorough (Ney et al., 1997; Chen and Goodman, 1999), have been carried out in terms of test set perplexities. As a consequence, none of them has analyzed the interdependence between the smoothing technique, probability scaling at decoding time and final system performance. The argument is that the linear correlation between test set perplexity and word error rate is very strong (Chen and Goodman, 1999). However, this correlation is dependent, once again, on the relationship being analyzed.

3. The Syntactic Language Model

A syntactic approach of the well-known *n*-gram models, *k*-Testable in the Strict Sense (*k*-TSS) LMs, has been used in this work to be integrated into a CSR system. The use of k-TSS regular grammars allows us to obtain a deterministic Stochastic Finite State Automaton (SFSA) integrating K k-TSS models (with k=1, 2 ... K) into a self-contained model (Varona and Torres, 1999; Torres and Varona, 2001).

Each Stochastic Finite State Automaton representing a k-gram model can be directly obtained from a set of training samples (García and Vidal, 1990). Such an automaton (k-TSS SFSA) is defined as the five-tuple $(\Sigma, Q^k, q_0, F, \delta^k)$ where:

- $\Sigma = \{\omega_j\}, j = 1 \ldots |\Sigma|$, is the vocabulary, that is the set of words appearing in the training corpus.
- Q^k is the set of states associated with the model of order k. Each state represents a string of $k-1$ words $\omega_{i-(k-1)} \ldots \omega_{i-1}$, where i stands for a generic index in any string $\omega_1 \ldots \omega_i \ldots$ appearing in the training corpus. Such a state is labeled as $\omega_{i-(k-1)}^{i-1}$.
- the automaton has a unique initial state $q_0 \in Q^k$, and F is the set of final states of the automaton and represents the final substrings of length $k-1$.
- δ^k is the transition function $\delta^k : Q^k \times \Sigma \rightarrow Q^k \times [0 \ldots 1]$. $\delta^k(q, \omega_i) = (q_d, P(\omega_i \mid q))$ defines a destination state $q_d \in Q^k$ and a probability $P(\omega_i \mid q) \in [0 \ldots 1]$ to be assigned to each element $(q, \omega_i) \in Q^k \times \Sigma$. Each transition represents a k-gram; it is labeled by its last word ω_i and connects two states labeled by $k-1$ words. When the k-gram $\omega_{i-(k-1)}\omega_{i-(k-1)+1} \ldots \omega_{i-1}\omega_i$ is observed, an outgoing transition from the first to the second state is set and labeled by ω_i; $P(\omega_i | \omega_{i-(k-1)}^{i-1})$ is the probability associated with the observed k-gram $\omega_{i-(k-1)}\omega_{i-(k-1)+1} \ldots \omega_{i-1}\omega_i$. Thus:

$$\delta^k\left(\omega_{i-(k-1)}^{i-1}, \omega_i\right) = \left(\omega_{i-(k-1)+1}^{i}, P\left(\omega_i \mid \omega_{i-(k-1)}^{i-1}\right)\right) \tag{2}$$

The model defined above is a deterministic, and hence unambiguous, stochastic finite state automaton (García and Vidal, 1990). Thus, the probability assigned to a sentence $\Omega \equiv \omega_1 \ldots \omega_l$ of length l, i.e. the probability of string Ω being accepted by the automaton is obtained as the product of the probabilities of the transitions used to accept Ω:

$$P(\Omega) = \prod_{i=1}^{l} P\left(\omega_i \mid \omega_{i-(k-1)}^{i-1}\right) \tag{3}$$

The unambiguity of the automaton also allows us to obtain a maximum likelihood estimation of the probability of each transition $\delta^k(\omega_{i-(k-1)}^{i-1}, \omega_i)$ as (Chandhuri and Booth, 1986):

$$P\left(\omega_i \mid \omega_{i-(k-1)}^{i-1}\right) = \frac{N\left(\omega_i \mid \omega_{i-(k-1)}^{i-1}\right)}{\sum_{\forall \omega_j \in \Sigma} N\left(\omega_j \mid \omega_{i-(k-1)}^{i-1}\right)} \tag{4}$$

where $N(\omega_i \mid \omega_{i-(k-1)}^{i-1})$ is the number of times the word ω_i appears at the end of the k-gram $\omega_{i-(k-1)} \ldots \omega_{i-1}\omega_i$, that is the count associated with the transition labeled by ω_i coming from the state labeled as $\omega_{i-(k-1)}^{i-1}$.

The k-TSS language model defined above can be considered as a syntactic version of an n-gram. Moreover, it has been shown (Dupont and Miclet, 1998) that the probability distribution obtained through an n-gram model is equivalent to the distribution obtained by a stochastic grammar generating k-TSS language, where k plays the same role as n does in n-grams. The main advantage of using automata to represent n-grams is that the structural features are explicitly developed in the formal language theory, and thus, explicitly included in the model (Hopcroft and Ullman, 1979; Torres and Varona, 2001). A complete formulation of this formalism can be found in Torres and Varona (2001).

4. Back-off Smoothing

The maximum likelihood estimation of the probability $P(\omega_i \mid \omega_1 \ldots \omega_{n-1})$ is

$$P_{ML}(\omega_i \mid h) = \frac{N(\omega_i \mid h)}{\sum_{\omega_j} N(\omega_j \mid h)} \tag{5}$$

where $h = (\omega_{i-(n-1)}^{i-1})$ is a history representing a sequence of $n-1$ words, $N(\omega_i \mid h)$ is the number of times the word ω_i appears after history h.

This estimation cannot deal with practical speech recognition applications where new sequences of words may be allowed, even if they do not appear (*unseen events*) in training corpora. The smoothing technique adjusts the maximum likelihood estimation in order to assign a probability to any sequence of words. A certain mass of probability is subtracted from *seen* events (sequences of words appearing at training time) and then redistributed among *unseen* events.

Smoothing algorithms use lower n-gram models to assign a probability to unseen n-grams. There are two basic approaches for combining n-grams of different order: linear interpolation (Jelinek and Mercer, 1980) and back-off (Katz, 1987). Interpolated models are obtained as linear or non-linear combinations of the probabilities of the model to be smoothed $P(\omega_i \mid h)$ and the probabilities of a more general model $P(\omega_i \mid h^*)$ in the following way:

$$P(\omega_i \mid h) = \lambda P(\omega_i \mid h) + (1-\lambda)P(\omega_i \mid h^*) \tag{6}$$

where h is a history representing a sequence of $n-1$ words and h^* represents a history of words shorter than h. In general, λ depends on the history h. λ is usually estimated using the forward-backward algorithm.

In back-off smoothing (Katz, 1987) the probability to be assigned to unseen n-grams is recursively obtained from less accurate models, i.e. $n-1, \ldots, 1$. Under this formalism, the probability $P(\omega_i|h)$ is estimated according to:

$$P(\omega_i|h) = \begin{cases} (1-\lambda)\dfrac{N(\omega_i|h)}{N(h)} & N(\omega_i|h) \neq 0 \\ \left(\displaystyle\sum_{\forall \omega_j / N(\omega_j|h)\neq 0} \lambda \frac{N(\omega_j|h)}{N(h)}\right) \dfrac{P(\omega_i|h^*)}{\sum_{\forall \omega_j N(\omega_j|h)=0} P(\omega_j|h^*)} & \\ & N(\omega_i|h) = 0 \end{cases} \tag{7}$$

where $N(\omega_i \mid h)$ is the number of times that word ω_i appears after history h, $N(h) = \sum_{\forall \omega / N(\omega_i \mid h)\neq 0} N(\omega_i \mid h)$ and $P(\omega_i \mid h^*)$ is the estimated probability associated with the same event in the more general model ($n-1$, $n-2, \ldots, 1$) and $(1-\lambda)$ represents the discount factor, that is, the amount of probability to be subtracted and then redistributed among unseen events.

The difference between interpolation and back-off matters fundamentally in the details of the mathematical models (Ney et al., 1997; Chen and Goodman, 1999). Moreover, a back-off version of an interpolated algorithm can be created in a straightforward way (Chen and Goodman, 1999). In this work back-off smoothing has been chosen because the recursive scheme involved has been better integrated into the finite state formalism used to build the language model. A full definition and implementation of this formalism can be found in Torres and Varona (2000) and Torres and Varona (2001).

The discount factor $(1-\lambda)$ has many different formulations (Ney et al., 1997; Chen and Goodman, 1999; Chen and Rosenfeld, 2000; Clarkson and Rosenfeld, 1997; Goodman, 2001) leading to a wide range of probability distributions, from very low to very high smoothed models. In high-smoothed LMs the probability reserved by the smoothing technique for the *unseen* events is bigger than in low-smoothed LM. As a consequence the gap between LM probabilities in high-smoothed LMs is smaller than in low-smoothed models. This variability is strongly related to the optimum scaling of LM probabilities required at decoding time to get good system performances. We now review classical smoothing discountings under this point of view. We first present techniques where the discount is applied to the whole set of seen events in the training corpus, as suggested in Eq. (7). We then deal with the Kneser-Ney discount, which is considered by many authors as the most reliable smoothing technique. Finally we deal with the Katz's back-off proposal where the discount is only applied over scarcely seen events. The new Delimited discounting, defined by the authors in previous works (Varona and Torres, 2000), is also included in this case.

4.1. Discounting Over all Seen Events

The most classical approaches are Witten-Bell, Add-one, Absolute and Linear discountings (Clarkson and Rosenfeld, 1997). In Absolute and Linear proposals the mass of probability $(1-\lambda)$ subtracted from seen events depends on certain parameter values that should be optimized. Thus, LMs with very different degrees of smoothing can be obtained.

Witten-Bell Discounting. In Witten-Bell, the discount $(1-\lambda)$ basically depends on the number of different events $T(h)$ following the history h. That is:

$$1-\lambda = \frac{N(h)}{N(h)+T(h)} \tag{8}$$

It is widely used since it leads to low test set perplexities when compared to other classical back-off methods

(Jurafsky and Martin, 2000; Bordel et al., 1994; Clarkson and Rosenfeld, 1997). However, a dependence has been found (Chen and Goodman, 1999) between perplexity and the size of the training of the LMs when Witten-Bell discounting is used.

In this case a quite important mass of probability is assigned to unseen events (high-smoothing) and the gap between seen and unseen probabilities is reduced. Combinations of words unseen in training can have a relatively high probability in tests.

Add-one Discounting. This is a very simple discounting method, adding one to all the counts. It is calculated as:

$$1 - \lambda = \frac{N(h)}{N(h) + 1} \tag{9}$$

This method does not usually perform well and thus is not commonly used by itself Jurafsky and Martin (2000). It is usually applied as part of more complex methods[2] (Ney et al., 1997)

Since $1 \leq T$, the mass of probability to be discounted and then redistributed among *unseen events* is smaller (low smoothing) when using Add-one discount then when using Witten-Bell one. The gap among LM probabilities is therefore bigger using Add-One discounting.

Absolute Discounting. This discounting schema (Ney et al., 1997) consists of subtracting a constant b from each count $N(\omega_i \mid h)$ in the following way:

$$1 - \lambda = \frac{N(\omega_i \mid h) - b}{N(\omega_i \mid h)} \tag{10}$$

Thus, $(1 - \lambda)$ depends on the value of the count $N(\omega_i \mid h)$. Parameter b regulates the degree of smoothing, that is the amount of probability discounted and then distributed. In general parameter b depends on the history h. High values of b lead to high smoothed models. The version included in the CMU toolkit (Clarkson and Rosenfeld, 1997) uses a previously established value of b (Ney et al., 1997) to get a discount similar to the Good-Turing[3] (Katz, 1987; Chen and Goodman, 1999) one: $b = n_1/(n_1 + 2n_2)$, where n_i is the number of events seen i times. However, this value is impracticable in many situations, such as when using cutoffs, where it needs to be rectified. Parameter b is experimentally optimized to get low perplexity LMs (Ney et al., 1997; Chen and Goodman, 1999). The Bounded discount (Ney et al., 1997) is a variant of the Absolute discount where high counts are not modified (Ney et al., 1997).

Linear Discounting. In this case a quantity proportional to each count is subtracted from the count itself in the following way:

$$1 - \lambda = (1 - l) \tag{11}$$

The value of the parameter l regulates the degree of smoothing, being the discounting factor independent of any count, history or vocabulary size. The value of l established (Ney et al., 1997) to get a discount similar to the Good-Turing discount (Katz, 1987) and used by the CMU toolkit (Clarkson and Rosenfeld, 1997) is $l = n_1/N(h)$. Once again, this value is impracticable in many situations.

4.2. *Kneser-Ney smoothing*

The Kneser-Ney smoothing is an extension of absolute smoothing introduced by Kneser and Ney (1995). In it, the discount is also applied over the whole set of *seen* events. The highest-order distribution is exactly the same as that obtained in absolute smoothing. But lower-order distributions are modified: the probability distribution $P(\omega_i \mid h^*)$ in Eq. (7) is substituted by distribution $\beta(\omega_i \mid h^*) = C(\omega_i \mid h^*)/C(h^*)$ where $C(\omega_i \mid h^*)$ is the count of the number of different contexts in which the sequence of words $\omega_{i-(n-2)}^{i}$ appears and $C(h^*) = \sum_{\forall \omega_j / C(\omega_j \mid b_q) \neq 0} C(\omega_j \mid b_q)$. As a consequence, the unigram probability is not proportional to the number of occurrences of a word, but to the number of different preceding words. Thus, Eq. (7) is now:

$$P(\omega_i \mid h) = \begin{cases} (1 - \lambda)\dfrac{N(\omega_i|h)}{N(h)} & N(\omega_i|h) \neq 0 \\ \left(\displaystyle\sum_{\forall \omega_j / N(\omega_j|h) \neq 0} \lambda \frac{N(\omega_j|h)}{N(h)} \right) \dfrac{\beta(\omega_i|h^*)}{\displaystyle\sum_{\forall \omega_j N(\omega_j|h) = 0} \beta(\omega_j|h^*)} & \\ & N(\omega_i|h) = 0 \end{cases} \tag{12}$$

where history $h \equiv \omega_{i-(n-1)}^{i-1}$ represents a sequence of $n - 1$ words and the shorter history $h^* \equiv \omega_{i-(n-2)}^{i-1}$ represents a sequence of $n - 2$ words.

Kneser-Ney smoothing really proposes to change distribution $P(\omega_i|h^*)$ by distribution $\beta(\omega_i|h^*)$ in Eq. (7). Then, any discount could be applied. In this work only the Absolute discount is used as it proposed in the relevant literature. Many authors have stated that Kneser-Ney smoothing outperforms all other smoothing techniques (Goodman, 2001) even with high-order n-grams. New variants of this smoothing have also been more recently proposed (Chen and Goodman, 1999).

4.3. Katz's Schema: Discounting Over Scarcely Seen Events

The well known Good-Turing formula (Katz, 1987) is based on the assumption that high counts are better estimated than low counts. Katz extended this idea to back-off smoothing by including combinations with more general probability distributions, i.e., with lower order models $n-1, n-2, \ldots, 1$. According to this formalism the probability associated with events occurring more than a fixed number of times, say r times, is estimated under a maximum likelihood criterion whereas events occurring less than r times, $N(\omega_i \mid h) \leq r$, are discounted a certain mass of probability (Katz, 1987). Thus the probability $P(\omega_i \mid h)$ is estimated according to:

$$P(\omega_i|h) = \begin{cases} \dfrac{N(\omega_i|h)}{N(h)} & N(\omega_i|h) \neq 0 \wedge N(\omega_i|h) > r \\ (1-\lambda)\dfrac{N(\omega_i|h)}{N(h)} & N(\omega_i|h) \neq 0 \wedge N(\omega_i|h) \leq r \\ \left(\displaystyle\sum_{\substack{\forall\omega_j / \\ N(\omega_j|h)\neq 0}} \lambda \frac{N(\omega_j|h)}{N(h)} \right) \dfrac{P(\omega_i|h^*)}{\displaystyle\sum_{\substack{\forall\omega_j / \\ N(\omega_j|h)=0}} P(\omega_j|h^*)} & N(\omega_i|h) = 0 \end{cases} \tag{13}$$

This schema has been applied along with alternative estimations for the discount due to the drawbacks of the Turing formula in practical applications. The final Katz proposal keeps the total probability assigned to *unseen* events proposed by the Good-Turing formula, i.e. $n_1/N(h)$, while estimating the discount as:

$$1-\lambda = \frac{\frac{(N(\omega_i|h)+1)nN(\omega_i|h)+1}{N(\omega_i|h)nN(\omega_i|h)} - \frac{(r+1)n_{r+1}}{n_1}}{1 - \frac{(r+1)n_{r+1}}{n_1}} \tag{14}$$

Delimited Discounting. In Turing discounting, the discounted mass of probability depends on $n_1, n_{2,} \ldots, n_{r+1}$, being n_i the number of events which occur i times. This approach puts some constraints to the relative values of $n_1, n_2, \ldots, n_{r+1}$ which are not always satisfied by n-gram models with medium and high values of n, due to the lack of an adequate distribution of the samples. To avoid these problems, Delimited discounting was proposed in Varona and Torres (2000). As in Katz's model (see Eq. (13)), the discounting operation is limited to low counts, i.e. $N(\omega_i|h) < r$, in the following way:

$$(1-\lambda) = d - \tau(r - N(\omega_i|h)) \quad \tau, d < 1 \wedge \tau <<< d \tag{15}$$

Discounting depends on d and τ parameter values, which must be less than one. When the count $N(\omega_i \mid h) \leq r$ is high the discount applied is low (low smoothing). The minimum discount is applied for $N(\omega_i \mid h) = r$ and only depends on parameter d. However, when $N(\omega_i \mid h) = 1$ the maximum discount, $(d - \tau(r-1))$, is applied.

Another problem to be addressed when using Katz' discounting is that additional checks are required for those histories for which all the events are *seen* more than r times (see Eq. (13)). The remedy used in the CMU toolkit is to increase the count $N(h)$ by one using the gained probability mass $\frac{1}{(N(h)+1)}$ to be redistributed over *unseen* events' probabilities, which does not agree with Katz's discounting philosophy. In Delimited discounting, only the minimum counts are decremented (discounting only depends on parameter d) in the following way:

$$P(\omega_i|h) = \begin{cases} \dfrac{N(\omega_i|h)}{N(h)} & N(\omega_i|h) \neq 0 \wedge N(\omega_i|h) > \min\limits_{\forall\omega_i}((N(\omega_i|h)) \\ d\dfrac{N(\omega_i|h)}{N(h)} & N(\omega_i|h) \neq 0 \wedge N(\omega_i|h) = \min\limits_{\forall\omega_i}((N(\omega_i|h)) \\ \left(\sum\limits_{\substack{\forall\omega_j/N(\omega_j|h)=0 \\ \wedge 1 \leq N(\omega_j|h) \leq r}} [1-d]\dfrac{N(\omega_j|h)}{N(h)} \right) \dfrac{P(\omega_i|h^*)}{\sum\limits_{\substack{\forall\omega_j/ \\ N(\omega_j|h)=0}} P(\omega_j|h^*)} & N(\omega_i|h) = 0 \end{cases} \tag{16}$$

In Delimited discounting the parameter d mainly regulates the amount of probability discounted and then distributed, that is, the degree of smoothing of the model.

5. Experimental Evaluation

Experimental evaluation was carried out with two Spanish databases of very different degrees of difficulty. The first, Bdgeo, is a task-oriented corpus representing a set of queries to a Spanish geography database (Díaz et al., 1998). It was designed to test integrated systems (acoustic, syntactic and semantic modeling) in automatic speech understanding and includes a vocabulary of 1208 words. Speech utterances were recorded in laboratory environments at 16 KHz. The second database, Info_tren, consisted of 227 Spanish dialogues on train information that were recorded by a consortium of Spanish research groups as part of a project to develop a human-machine dialogue system (Rodríguez et al., 2001a). The dialogues were uttered in a spontaneous way and thus included acoustic, lexical and syntactic disfluencies. They were recorded at 8 KHz across telephone lines applying the well known Wizard of Oz mechanism. Info_tren task (Bonafonte et al., 2000) has a vocabulary of around 2000 words plus 15 different acoustic types of disfluencies including noises, filled pauses, lengthenings, etc. (Rodríguez et al., 2001).

In these experiments, the smoothing and discounting techniques reviewed in Section 4 were applied to each LM to be evaluated. The final probability distribution, and thus the degree of smoothing, is regulated by a parameter in Absolute (Ad), Linear (Ld), Kneser-Ney (KNd) and Delimited (Dd) discounts. Parameters b, l, b and d respectively (see Section 4), are optimized to get the lowest perplexity values for each database. Then two more values leading to a lower and a higher smoothed Language Model are also considered for each smoothing technique. As a consequence, a wide range of probability distributions, from very low to very high smoothed language models, is evaluated. Finally, Witten-Bell (WBd) and Add-one (AOd) discountings are also considered in the experiments. No parameter regulates the degree of smoothing in these cases. However, a smaller mass of probability is redistributed among unseen events when using Add-one than when using Witten-Bell discounting.

In a first series of experiments (Section 5.1) the evaluation is carried out in terms of the test set perplexity (PP). In the second series of experiments (Section 5.2) the language models obtained are directly integrated into the CSR system (Rodríguez et al., 2000). System performances are measured in terms of both the Word Error Rate and the Average number of Active Nodes in the trellis needed to decode a sentence. Finally, in a third series of experiments (Section 5.3), LM probabilities are modified by introducing an exponential scaling factor $P(\Omega)^\alpha$ (Jelinek, 1996). The object of these experiments is to get the lowest WER by selecting the adequate value of α. These experiments show that this optimization is also dependent of the smoothing technique and is not independent of the task and available training data.

5.1. Evaluating the Test Set Perplexity

The evaluation of an LM and its suitability to each application task should be carried out in terms of final system performance. However they are usually evaluated in terms of perplexity. The test set Perplexity (PP) is based on the mean log probability that a LM assigns

Table 1. Perplexity (PP) evaluation of k-TSS LMs with $k = 2 \ldots 6$ for the Bdgeo task. Parameters regulating the degree of smoothing are optimized to get optimum PP. Then a lower and a higher smoothed model are also considered. AOd discounting can be considered a lower smoothing version of WBd.

		Absolute (Ad)			Linear (Ld)		
k	Witten-Bell (WBd)	Low smoothing $b = 0.01$	Optimum smoothing $b = 0.4$	High smoothing $b = 0.9$	Low smoothing $l = 0.01$	Optimum smoothing $l = 0.1$	High smoothing $l = 0.3$
2	13.10	13.74	12.87	13.45	13.24	13.72	15.06
3	7.54	**9.38**	7.42	8.21	**8.47**	7.79	9.22
4	**7.17**	10.01	6.86	**8.07**	9.01	**7.29**	8.22
5	7.22	12.52	**6.84**	8.49	10.44	7.45	8.02
6	7.37	14.01	6.85	8.86	11.60	7.66	**7.97**
		Kneser-Ney (KNd)			Delimited (Dd)		
k	Add-One (AOd)	Low smoothing $b = 0.01$	Optimum smoothing $b = 0.3$	High smoothing $b = 0.8$	Low smoothing $d = 0.99$	Optimum smoothing $d = 0.7$	High smoothing $d = 0.4$
2	13.89	13.65	12.78	13.12	13.41	13.01	14.11
3	8.30	**10.75**	**8.87**	**10.61**	**8.76**	7.60	**9.23**
4	**7.72**	13.22	9.44	13.27	9.41	**7.01**	9.29
5	8.15	16.54	10.28	16.65	11.08	7.02	9.73
6	8.43	18.10	11.62	18.23	12.31	7.13	10.20

Table 2. Perplexity (PP) evaluation of k-TSS LMs with $k = 2 \ldots 6$ for the Info_tren task. Parameters regulating the degree of smoothing are optimized to get optimum PP. Then a lower and a higher smoothed model are also considered. AOd discounting can be considered a lower smoothing version of WBd.

		Absolute (Ad)			Linear (Ld)		
k	Witten-Bell (WBd)	Low smoothing $b = 0.1$	Optimum smoothing $b = 0.7$	High smoothing $b = 0.9$	Low smoothing $l = 0.1$	Optimum smoothing $l = 0.3$	High smoothing $l = 0.5$
2	36.84	**46.54**	37.08	38.86	**39.29**	41.74	53.49
3	**34.88**	63.79	**35.14**	**38.07**	48.08	**40.17**	**46.58**
4	36.37	80.76	36.23	40.06	59.23	43.29	47.54
5	36.83	86.96	36.46	40.48	63.58	44.47	47.97
6	36.89	87.89	36.53	40.68	64.31	44.61	47.98
		Kneser-Ney (KNd)			Delimited (Dd)		
k	Add-One (AOd)	Low smoothing $b = 0.1$	Optimum smoothing $b = 0.6$	High smoothing $b = 0.8$	Low smoothing $d = 0.9$	Optimum smoothing $d = 0.4$	High smoothing $d = 0.2$
2	**57.22**	**43.86**	**35.13**	**37.12**	**44.63**	37.04	**43.78**
3	69.87	69.12	43.15	55.88	62.11	**35.02**	45.78
4	77.73	91.34	50.80	77.52	78.59	37.15	47.81
5	98.15	100.74	55.32	93.13	86.76	38.42	48.27
6	118.15	123.17	61.10	110.12	85.7	39.86	48.3

to a test set ω_1^L of size L, so that it is based exclusively on the probability of words which actually occur in the test as follows:

$$PP = P\left(\omega_1^L\right)^{-1/L} = e^{-\frac{1}{L}\sum_{i=1}^{L}\log(P(\omega_i|\omega_1^{i-1}))} \quad (17)$$

The test set perplexity measures the branching factor associated with a task, which depends on the number of different words in the text. Low perplexity values are obtained when high probabilities are assigned to the test set events by the LM being evaluated. Therefore, the relationship with acoustic models in a CSR system is not taken into account.

For this series of experiments, the Bdgeo training corpus used to obtain the language models consisted of 9150 sentences, including 82,000 words. The test set consisted of 200 different sentences. These sentences were then uttered by 12 speakers resulting in a total of 600 sentences that composed the speech test set to be used in the next series of experiments (see Sections 5.2 and 5.3). On the other hand, the training test for the Info_tren application task consists of the transcription of 1349 user turns. They correspond to 191 dialogues uttered by 63 speakers and result in 16500 words and 5000 disfluencies. The test set consisted of the transcription of 36 dialogues (308 user turns) corresponding to 12 new speakers (4000 words plus around 500 disfluencies). The corresponding utterances composed the speech test to be used to evaluate CSR system performance (Sections 5.2 and 5.3).

Tables 1 and 2 show the results of the evaluation of smoothed k-TSS Language Models with $k = 2 \ldots 6$, in terms of test set perplexity (PP), over Bdgeo (Table 1) and Info_tren (Table 2) application tasks.

For the Bdgeo application task, Table 1 shows that the Kneser-Ney (KNd) smoothing technique obtains optimum PP values with $k = 3$, Witten-Bell (WBd), Linear (Ld), Add-One (AOd) and Delimited discounts obtain optimum PP for $k = 4$ and Absolute discount (Ad) performs better for $k = 5$. The lowest PP (6.84) for the Bdgeo task is obtained by a 5-TSS language model smoothed by an Absolute (Ad) discount. However, differences around optimum values are not very significant.

Table 2 shows the evaluation through the Info_tren database. In this case, disfluencies are included in the vocabulary. LMs are trained and tested with different sets of transcribed spontaneous dialogues and thus the mismatch between training and test is quite high (Rodríguez et al., 2001): there are a high number of sequences of k words in the test not appearing in the training set. As a consequence, the Perplexity values associated with this task are quite high (see Table 2) especially for high values of k. Table 2 also shows major differences among perplexity values when different smoothing techniques are applied. In this case, the Add-One (AOd) and Kneser-Ney (KNd) discounting techniques obtain optimum PP values with $k = 2$ whereas Witten-Bell (WBd), Absolute (Ad), Linear (Ld) and Delimited (Dd) obtain optimum PP values for $k = 3$ models. The lowest PP (34.88) for the Info_tren task is obtained by a 3-TSS language model smoothed by Witten-Bell (WBd) discount.

Results of the experiment shown in Tables 1 and 2 are plotted on the left side of Figs. 1 and 2 respectively. The pictures on the right side only plot a trace per smoothing technique which corresponds to the optimum perplexity. Figure 1 shows that low-smoothing techniques show a degradation of PP values for $k > 3$ models in the Bdgeo task. The number of *seen events*, i.e. k-grams, appearing in both the training and test sets quickly decreases as k increases. Thus, a big mass of probability (high smoothing) needs to be redistributed among *unseen events* to get good perplexity behavior. However, the picture on the right side of Fig. 1 shows that for optimized smoothing PP remains quite consistent as k increases. This picture also shows higher PP values when the Kneser-Ney (KNd) discount is applied over $k > 2$ models.

Figure 2 shows a similar behavior of the smoothing techniques with k-TSS language models for the Info_tren database. However, $k = 3$ models do not perform much better than $k = 2$ models due to the lack of training data for this difficult task. The right side picture confirms a more consistent behavior of PP for high values of k when optimized smoothed models are applied. However, this picture shows higher PP values when Add-One (AOd) and Kneser-Ney (KNd) discounts are applied over $k > 2$ models.

5.2. *Integrating the Smoothed Models into the CSR System*

In the second series of experiments the language models evaluated in previous Section are directly integrated into the CSR system (Rodríguez et al., 2000). The uttered sentences of each test set are decoded by the time-synchronous Viterbi algorithm with a fixed beam-search to reduce the computational cost (Varona

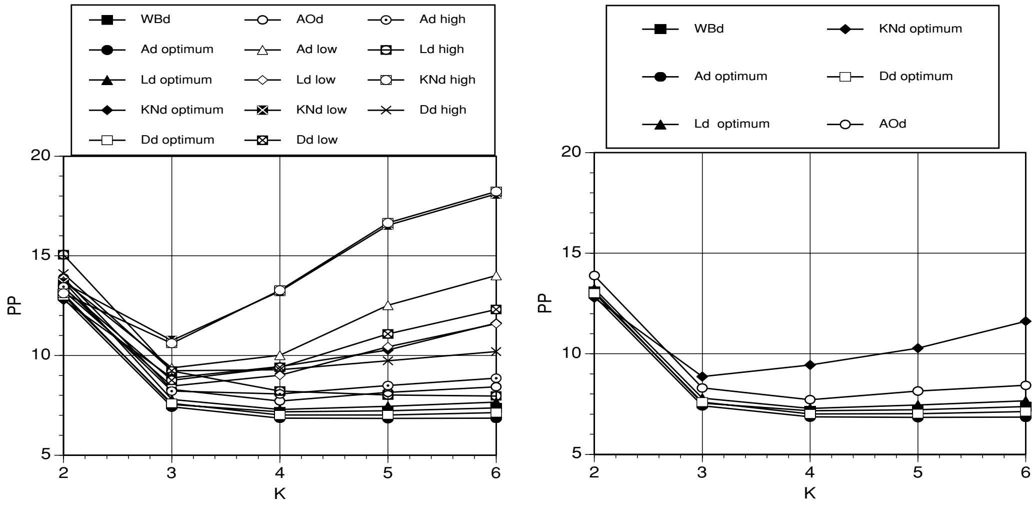

Figure 1. Perplexity (PP) evaluation of k-TSS LMs with $k = 2 \ldots 6$ for Bdgeo. The picture on the left side plots results in Table 1. That on the right side only plots a trace per smoothing technique which corresponds to optimum PP.

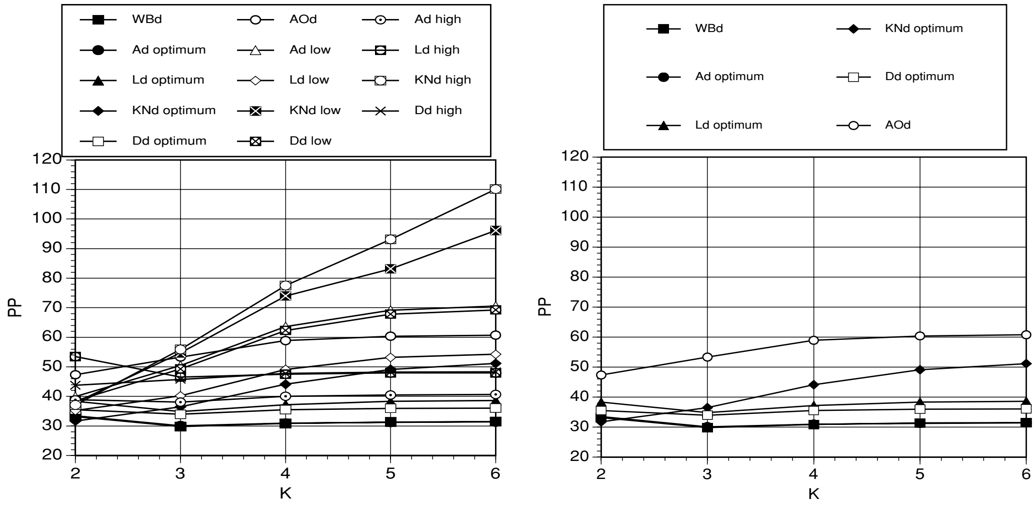

Figure 2. Perplexity (PP) evaluation of k-TSS LMs with $k = 2 \ldots 6$ for Info_tren. The picture on the left side plots results in Table 2. That on the right side only plots a trace per smoothing technique which corresponds to optimum PP.

and Torres, 2000). A chain of Hidden Markov Models (HMM) representing the acoustic model of the word phonetic chain replaces each transition of the k-TSS automaton. System performances are measured in terms of both the Word Error Rate (WER) and the Average number of Active Nodes (ANN) in the trellis (including acoustic and LM states) needed to decode a sentence. For these experiments, the LM probabilities are not modified when they are integrated into the CSR system, i.e. the exponential scaling factor α is set to 1 in $P(\Omega)^{\alpha}$.

Figures 3 and 4 show the WER and AAN values obtained through this series of experiments for the Bdgeo and Info_tren databases respectively. The PP evaluation of these LM is represented on the left side of Figs. 1 and 2, respectively.

Figures 3 and 4 show that low smoothing techniques obtain the best WER in both databases. As mentioned in Section 2, the gap among accumulated probabilities is usually bigger than the gap among LM probabilities when the LM is directly integrated into the CSR system ($\alpha = 1$). In that case, LM probabilities are often

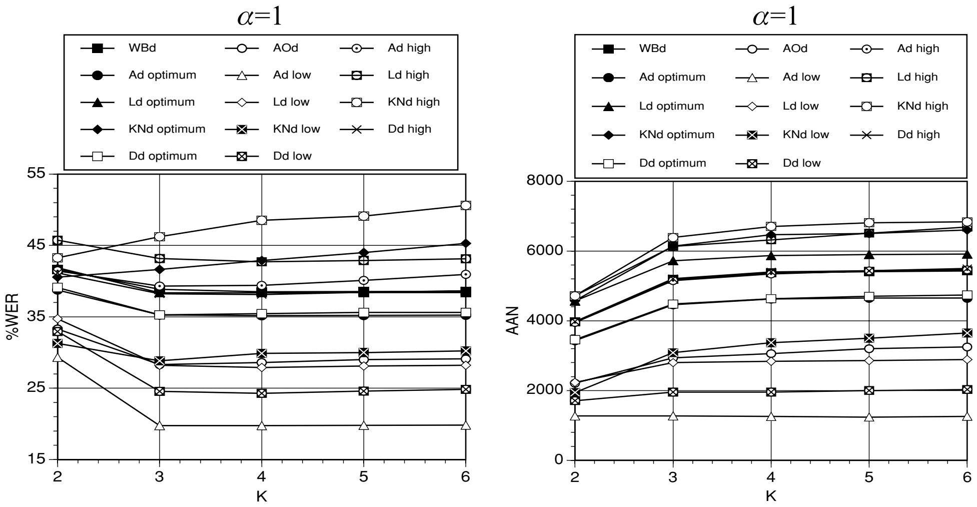

Figure 3. WER and AAN obtained for Bdgeo when k-TSS LMs with $k = 2 \ldots 6$ are directly (no scaling of LM probabilities: $\alpha = 1$) integrated into the CSR system. The PP evaluation of these models is shown on the left side of Fig. 1.

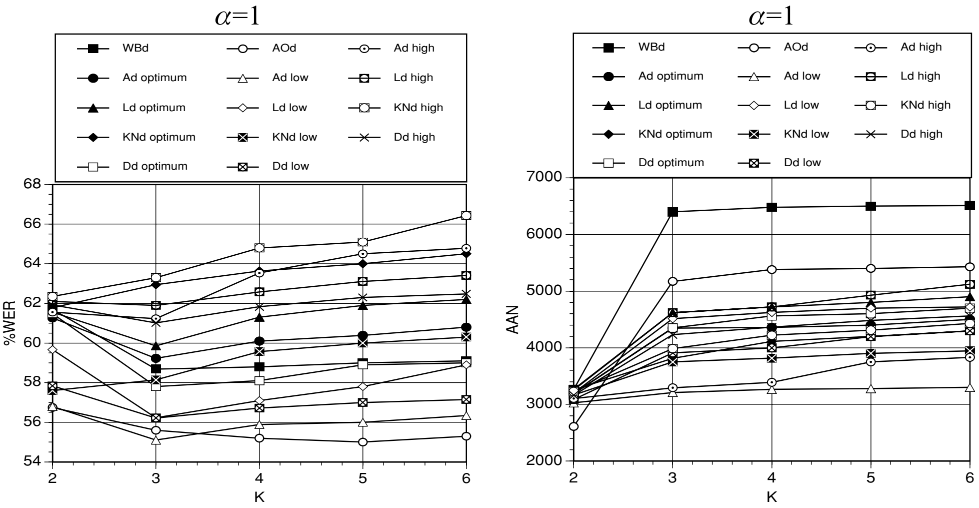

Figure 4. WER and AAN obtained for Info_tren when k-TSS LMs with $k = 2 \ldots 6$ are directly (no scaling of LM probabilities: $\alpha = 1$) integrated into the CSR system. The PP evaluation of these models is shown on the left side of Fig. 2.

irrelevant in the maximization procedure involved in the Viterbi algorithm. However, in low smoothed language models the gap among LM probabilities is bigger than in high smoothed ones and, as a consequence, LM probabilities are more competitive and relevant in the trellis. Thus, low smoothed LMs lead to lower WER when they are directly integrated into the CSR system ($\alpha = 1$) than high smoothed ones.

Pictures on the right side of Figs. 3 and 4 show that AAN behaves similarly WER: low smoothing LMs lead to low WER and to low AAN in the lattice. The beam-search needs to keep a lower number of active paths in the lattice when the LM probabilities become significant, (low smoothing techniques) resulting in lower computational costs. Moreover, the number of AAN does not increase as k does, even if the size of the language model increases (Torres and Varona, 2001).

Figures 5 and 6 only plot a trace per smoothing technique which corresponds to the best WER for the Bdgeo and Info_tren databases respectively.

Figures 5 and 6 show that models that optimize the test set perplexity (see Figs. 1 and 2) do not lead to

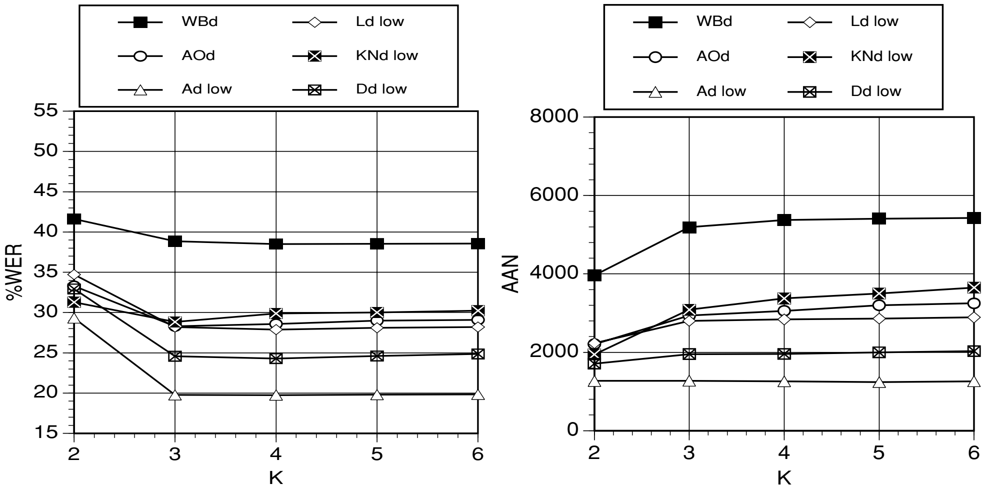

Figure 5. Selection of best WER for Bdgeo from Fig. 3: only a trace per smoothing technique. The PP evaluation of these models is shown on the right side of Fig. 1.

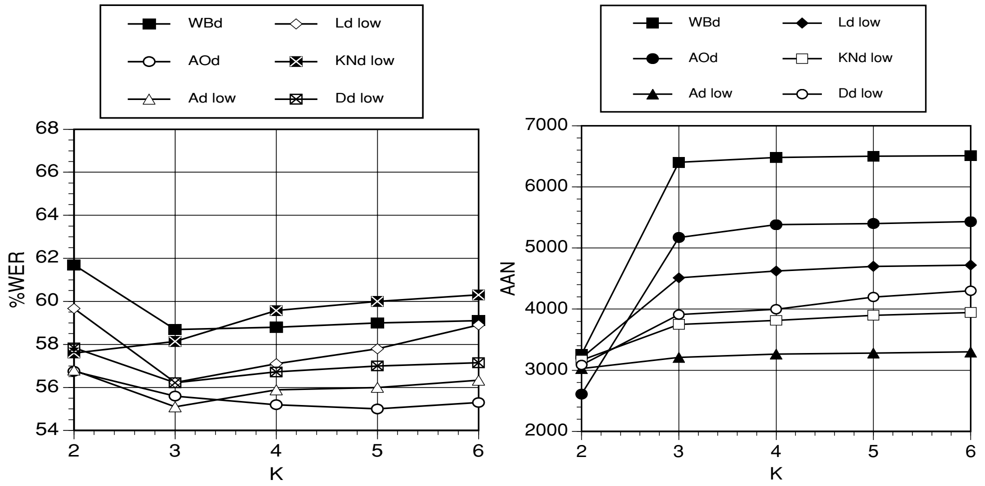

Figure 6. Selection of best WER for Info_tren from Fig. 4: only a trace per smoothing technique. The PP evaluation of these models is shown on the right side of Fig. 2.

the best system performances. In the Bdgeo task, the Absolute discount (WER = 19.76% for the $k = 4$ model), followed by Delimited discount, achieved the best system performances. In the Info_tren database the Add-One discount (WER = 55.38% for the $k = 5$ model) leads to the best results. Moreover, the best PP is obtained by $k = 3$ models in this database.

5.3. *Optimizing the System Performance*

LM probabilities are modified in the third series of experiments by introducing an exponential scaling factor $P(\Omega)^{\alpha}$ (Jelinek, 1996). Our main goal is to optimize the final system performance by selecting the right value of α. Tables 3 and 4 show the WER and AAN obtained for the Bdgeo and Info_tren databases respectively, when k-TSS LM's with $k = 2, \ldots, 4$ are integrated into the CSR system. Different values of the scaling exponential parameter ($\alpha = 1 \ldots 7$) are evaluated. Optimum performances are emphasized and underlined. Then, for $k = 5$ and $k = 6$ only model system performances corresponding to the $\alpha = 1$ (no scaling of LM probabilities) and to the value of α minimizing ($\alpha = optimum$) the WER are included.

Table 3. WER and AAN obtained for Bdgeo through k-TSS LMs with $k = 2 \dots 6$. Note that parameters regulating the degree of smoothing are optimized to get optimum PP; then a lower and a higher smoothed model are also considered (see Table 1 for PP evaluation). Scaling parameter values $\alpha = 1, \dots, 7$ are tested.

			Absolute (Ad)						Linear (Ld)					
	Witten-Bell (WBd)		Low smoothing $b = 0.01$		Optimum smoothing $b = 0.4$		High smoothing $b = 0.9$		Low smoothing $l = 0.01$		Optimum smoothing $l = 0.1$		High smoothing $l = 0.3$	
α	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN
1	41.62	3964	29.33	1270	38.76	3430	41.44	3950	34.73	2232	41.88	4565	45.72	4713
2	25.80	2588	19.29	813	24.22	2061	25.65	2544	22.26	1488	26.69	3192	30.64	3464
3	20.22	1508	16.1	533	18.54	1145	20.02	1449	17.34	923	20.9	1976	24.52	2335
4	16.99	764	**14.54**	**359**	16.00	591	17.06	701	15.44	509	17.95	996	21.13	1339
5	**15.80**	**380**	15.29	240	**15.17**	**306**	16.25	338	**14.63**	**284**	16.70	469	19.85	683
6	15.95	218	16.77	168	15.28	185	**16.23**	**192**	15.87	180	**16.66**	**253**	**17.85**	**357**
7	17.01	143	18.81	124	17.36	127	18.45	128	17.99	127	17.29	159	19.10	212
1	38.85	5189	19.78	1273	35.28	4462	39.31	5152	28.21	2801	38.39	5723	43.17	6136
2	21.86	2984	11.96	610	19.34	2274	22.17	2939	15.67	1442	21.35	3472	26.24	4172
3	15.35	1529	**10.69**	**347**	13.34	1084	15.41	1487	11.25	728	15.24	1890	19.22	2599
4	11.74	702	12.11	226	10.54	499	12.00	659	**10.50**	**349**	12.63	859	15.71	1382
5	**10.82**	**328**	13.93	146	**10.30**	**241**	**11.61**	**305**	11.63	182	**11.24**	**388**	13.78	654
6	10.85	179	18.2	102	11.56	140	12.17	166	14.98	114	11.72	199	**13.04**	**333**
7	13.04	114	21.88	77	14.13	95	14.67	107	18.4	82	13.15	119	13.49	190
1	38.50	5374	19.76	1258	35.16	4624	39.39	5344	27.89	2838	38.37	5868	42.72	118
2	21.86	3053	12.57	566	18.57	2316	21.59	3036	14.63	1376	20.76	3464	25.50	6326
3	14.44	1544	**12.06**	**313**	12.76	1083	15.32	1534	11.33	670	14.37	1827	18.05	4248
4	10.92	704	13.75	202	10.22	492	12.55	683	**11.13**	**312**	11.92	812	14.70	2606
5	10.24	328	17.09	129	**9.49**	**236**	**11.93**	**317**	13.24	161	**10.84**	**363**	12.59	1363
6	**10.22**	**177**	21.38	89	11.11	137	12.72	173	18.18	100	11.03	183	**12.36**	**637**
7	12.48	113	24.94	67	13.84	92	14.84	111	22.1	72	13.38	109	12.22	323
1	38.53	5410	19.81	1242	35.21	4656	40.10	5430	28.17	2860	38.39	5910	42.90	7213
op	10.24	212	11.83	325	10.02	240	12.17	397	13.11	345	11.02	381	13.98	721
1	38.56	5432	19.83	1265	35.23	5673	40.95	5461	28.19	2893	38.40	5936	43.13	7287
op	10.51	218	12.10	247	10.10	243	12.21	415	14.56	361	11.21	392	14.01	736

			Kneser-Ney (KNd)						Delimited (Dd)					
	Add-One (AOd)		Low smoothing $b = 0.01$		Optimum smoothing $b = 0.3$		High smoothing $b = 0.8$		Low smoothing $d = 0.99$		Optimum smoothing $d = 0.7$		High smoothing $d = 0.4$	
α	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN
1	33.29	2209	31.28	1944	40.54	4575	43.27	4714	32.93	1709	39.11	3458	41.14	3979
2	21.60	1207	19.76	1003	24.86	3120	27.05	3374	21.01	1045	24.06	2088	25.37	2581
3	17.33	684	16.28	575	18.63	1840	20.67	2156	16.37	645	18.58	1156	19.38	1478
4	**14.98**	**416**	**14.57**	**366**	15.94	855	16.87	1086	**14.71**	**397**	16.03	592	16.92	714
5	15.20	258	15.32	240	**14.59**	**366**	**15.79**	**494**	15.05	249	**15.33**	**305**	**15.69**	**338**
6	15.93	173	16.79	168	15.48	186	15.94	218	16.47	169	15.73	185	16.24	191
7	18.14	126	18.85	124	17.76	124	17.30	128	18.64	123	15.95	127	17.90	125

(*Continue on next page*)

Table 3. (*Continued.*)

			Kneser-Ney (KNd)						Delimited (Dd)					
	Add-One (AOd)		Low smoothing $b = 0.01$		Optimum smoothing $b = 0.3$		High smoothing $b = 0.8$		Low smoothing $d = 0.99$		Optimum smoothing $d = 0.7$		High smoothing $d = 0.4$	
α	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN
1	28.3	2935	28.83	4085	41.64	6139	46.23	6387	24.57	1954	35.28	4479	38.22	5209
2	16.49	1325	14.68	1938	23.76	4093	28.62	4586	13.33	879	19.39	2292	21.46	3024
3	12.5	633	**11.52**	**969**	15.79	2496	19.69	3058	**10.83**	**454**	13.3	1088	15.16	1551
4	**10.98**	**339**	11.69	416	12.39	1423	15.76	1940	11.14	258	10.64	498	11.84	693
5	11.04	193	14.01	251	**11.30**	**671**	13.44	1081	12.72	157	**10.13**	**240**	**10.65**	**315**
6	13.08	123	16.80	185	11.40	352	**12.84**	**518**	15.65	106	12.31	140	12.19	173
7	15.67	88	22.59	147	13.34	212	13.87	259	19.6	79	14.39	95	14.34	110
1	28.59	3058	29.89	4373	42.89	6467	48.53	6703	24.28	1957	35.43	4634	38.11	5403
2	16.03	1356	15.80	2097	25.11	4402	31.37	4933	13.22	825	18.63	2326	20.87	3129
3	11.91	640	13.02	1042	16.72	2756	22.82	3411	**11.06**	**410**	12.46	1083	14.30	1610
4	10.89	339	**12.93**	**437**	13.31	1616	19.27	2260	11.52	229	10.13	490	11.19	726
5	**10.67**	**190**	16.34	250	**12.89**	**794**	16.80	1338	15.13	139	**9.74**	**235**	**10.53**	**335**
6	13.44	120	21.60	180	12.96	415	**16.17**	**686**	19.38	93	11.79	136	11.87	184
7	10.89	85	27.64	140	14.73	243	17.28	348	23.08	69	14.18	92	14.63	118
1	29.10	3200	30.02	3512	44.00	5530	49.10	6810	24.61	2013	35.61	4713	38.45	5432
op	11.03	213	13.51	451	13.31	802	17.90	723	11.56	432	10.01	240	10.91	345
1	29.23	3249	30.23	3658	45.31	5610	50.61	6835	24.85	2038	35.62	4747	38.64	5497
op	11.21	125	14.05	468	14.17	867	18.63	785	112.38	456	10.51	256	10.98	360

Tables 3 and 4 show that, for any k-TSS model, major decreases in word error rates (down to a minimum) can be observed when the balance parameter α is increased.

Tables 3 and 4 also show that low smoothing techniques perform better when α has not reached its optimum value ($\alpha < optimum$). Moreover, they need a lower scaling of LM probabilities, i.e. a lower value of α, to get the optimum WER (bold typed) than PP optimized and high-smoothing techniques. Low smoothing techniques lead to a bigger gap among the LM probabilities than high smoothing ones, and therefore the relative weight of LM probabilities in the trellis is higher. In fact, the exponential scaling of LM probabilities leads to a probability redistribution, as any smoothing technique does. Thus, LM scaling can be considered as a new smoothing (exponential) applied to the previously smoothed LM probabilities needed to get optimum CSR performances. As a consequence, the effect of the smoothing technique in final system performance should be analyzed along with the effect of the exponential scaling of LM probabilities. In fact, there is a strong dependence between the smoothing technique and the value of the scaling parameter α needed to get the best performance from the system, which is, indeed, perplexity independent in many cases.

As the value of α is increases (up to the optimum value), the differences among performances are reduced. When optimum values of α are reached, similar optimum performances (WER and AAN) are obtained for all smoothing techniques evaluated. This behavior can be analyzed in Figures 7 and 8. They plot the CSR system performances (WER and AAN) achieved through all smoothed LM analyzed and optimum LM scaling (emphasized values in Tables 3 and 4), for the Bdgeo and Info_tren task, respectively.

Figure 7 shows an important decrease of WER for $k > 2$ and optimum WER for $k = 3$ and $k = 4$ models in the Bdgeo task. It also shows a quite constant computational cost (AAN) for any k-TSS model and most smoothing techniques.

For the Info_tren task, Fig. 8 shows a quite similar optimum WER for any smoothing technique, in spite of the great perplexity differences shown in Table 2 and Fig. 2. In this case, Perplexity seems not to be the most adequate parameter for evaluating the quality of a smoothing technique when the LM is designed to be integrated into a CSR system. Figure 8 also shows that the AAN needed to decode a sentence is twice as much for $k > 2$ models as for $k = 2$ whereas the WER remains nearly constant. For this database, *Table 4*. WER and AAN obtained for Info_tren through k-TSS LMs with $k = 2 \ldots 6$. Note that parameters regulating the degree of smoothing are optimized to get optimum PP; then a lower and a higher smoothed model are also considered (see Table 2 for PP evaluation). Scaling parameter values $\alpha = 1, \ldots, 7$ are tested.

			Absolute (Ad)						Linear (Ld)					
	Witten-Bell (WBd)		Low smoothing $b = 0.1$		Optimum smoothing $b = 0.7$		High smoothing $b = 0.9$		Low smoothing $l = 0.1$		Optimum smoothing $l = 0.3$		High smoothing $l = 0.5$	
α	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN
1	61.69	3260	56.80	3027	61.25	3186	61.56	3105	59.67	3202	61.56	3254	62.10	3266
2	50.23	2594	45.87	2202	48.78	2466	49.29	2364	48.35	2500	50.15	2614	50.71	2639
3	43.83	1912	41.06	1440	43.04	1743	43.37	1649	42.81	1800	43.94	1956	44.87	1988
4	41.08	1291	39.05	866	40.23	1122	40.85	1065	40.06	1182	41.45	1348	42.43	1379
5	**39.60**	**799**	**38.60**	**504**	38.81	667	40.46	659	**38.66**	**714**	40.15	854	41.62	877
6	40.32	467	40.58	331	**38.68**	**401**	**40.14**	**402**	39.69	420	**40.05**	**500**	**41.11**	**506**
7	41.75	294	42.46	237	40.99	251	42.21	250	41.49	271	41.56	306	43.23	301
1	58.69	6400	55.10	3210	59.23	3345	61.23	3295	56.23	4512	59.86	4620	61.90	4621
2	48.72	4668	43.27	2511	48.10	2642	54.32	2583	45.30	3987	47.99	4005	54.60	4015
3	42.14	3172	38.93	1883	45.76	1960	46.51	2015	43.37	2560	45.85	2615	46.79	2695
4	38.72	1978	**37.89**	**993**	41.45	1210	39.30	1681	**37.55**	**1547**	42.42	1675	43.89	1743
5	**38.01**	**1135**	40.10	534	**37.99**	**995**	**38.36**	**994**	37.69	830	**37.79**	**1150**	**38.82**	**1219**
6	38.41	631	44.67	334	38.02	568	39.33	579	39.94	464	38.79	641	39.02	676
7	41.58	378	49.83	231	41.13	338	41.41	344	44.71	289	40.70	375	41.44	386
1	58.80	6480	55.89	3265	60.10	3360	63.53	3390	57.10	4623	61.31	4720	62.58	4721
2	48.90	4720	47.01	2610	50.60	2683	57.30	2615	46.46	4001	49.15	4110	54.99	4113
3	42.25	3286	43.27	1943	53.21	2054	46.86	2140	43.70	2715	45.71	2762	46.52	2785
4	38.83	1815	**38.93**	**1061**	45.87	1355	43.80	1980	**38.56**	**2287**	43.34	1823	44.02	1940
5	**37.84**	**1269**	41.98	555	42.19	980	39.69	1135	38.86	1649	38.94	1237	38.72	1318
6	38.63	702	48.80	338	**38.82**	**655**	**38.86**	**664**	43.50	865	**38.00**	**677**	**38.51**	**723**
7	42.31	415	54.17	225	40.87	388	42.68	396	50.94	472	42.45	391	41.37	407
1	59.03	6512	56.03	3284	60.39	4411	64.50	3750	57.83	4723	61.91	4823	63.11	4930
op	38.01	1301	39.00	1115	39.03	702	38.91	698	38.91	2311	38.12	1438	39.90	741
1	59.17	6538	56.34	3312	60.85	4523	64.78	6831	58.91	4761	62.28	4915	63.41	5120
op	38.19	1340	39.17	1213	39.21	748	39.19	762	39.15	2348	38.35	1356	40.10	764

			Kneser-Ney (KNd)						Delimited (Dd)					
	Add-One (AOd)		Low smoothing $b = 0.1$		Optimum smoothing $b = 0.6$		High smoothing $b = 0.8$		Low smoothing $d = 0.9$		Optimum smoothing $d = 0.4$		High smoothing $d = 0.2$	
α	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN
1	56.75	2610	57.60	3164	61.78	3256	62.35	3227	57.84	3087	61.48	3191	61.92	3162
2	47.15	1827	47.14	2398	49.67	2594	50.20	2540	46.78	2295	49.24	2473	50.49	2422
3	42.13	1199	41.07	1641	43.22	1906	44.47	1833	40.95	1539	43.51	1751	44.49	1686
4	**39.89**	**760**	**39.16**	**1011**	40.32	1279	41.52	1211	39.03	940	41.08	1129	42.46	1069
5	40.75	484	39.45	563	**38.73**	**778**	40.40	733	**38.54**	**540**	38.86	665	41.84	630
6	42.30	335	40.97	338	39.14	440	**40.19**	**445**	40.37	341	**38.57**	**404**	**41.35**	**398**
7	43.92	245	43.75	237	41.48	260	42.50	286	42.72	239	42.01	253	43.63	250

(*Continued to next page.*)

Table 4. (*Continued.*)

			Kneser-Ney (KNd)						Delimited (Dd)					
	Add-One (AOd)		Low smoothing $b = 0.1$		Optimum smoothing $b = 0.6$		High smoothing $b = 0.8$		Low smoothing $d = 0.9$		Optimum smoothing $d = 0.4$		High smoothing $d = 0.2$	
α	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN	WER	AAN
1	55.60	5172	58.14	3750	62.94	3813	63.30	3986	56.22	3911	57.80	4351	61.02	4230
2	45.21	3233	45.80	2713	50.53	2998	51.12	3126	43.84	3213	44.30	3983	50.13	3710
3	41.50	1876	40.33	1973	43.76	2113	45.13	2250	39.78	2314	40.62	3001	44.53	2921
4	**39.36**	**1060**	**38.42**	**1457**	39.66	1915	42.10	1920	**38.38**	**1097**	37.91	1740	41.34	1666
5	40.24	610	41.03	806	**38.51**	**1448**	**41.78**	**1346**	39.48	579	**38.16**	**979**	**40.32**	**955**
6	43.13	386	44.95	484	40.06	898	42.49	846	44.39	349	39.12	569	41.53	579
7	47.41	269	50.87	312	41.15	553	45.36	538	49.06	236	43.72	340	44.26	350
1	55.20	5380	59.57	3816	63.64	4111	64.80	4221	56.72	3998	58.10	4561	61.83	4365
2	45.00	3410	46.52	2803	51.58	3214	53.31	3413	47.72	3345	52.01	4180	53.47	3867
3	41.04	2237	40.05	2111	45.66	2365	47.42	2462	43.57	2413	46.83	3943	48.03	3010
4	**39.10**	**1250**	**39.87**	**1651**	41.21	1965	45.13	2102	**38.89**	**1170**	44.58	2564	46.38	2501
5	41.24	708	43.27	898	**39.68**	**1364**	**43.58**	**1374**	41.18	603	**38.88**	**1114**	**40.95**	**1090**
6	43.70	436	48.04	517	40.71	1058	45.91	970	47.42	354	39.34	643	42.31	658
7	48.26	296	55.53	322	42.87	646	48.14	618	54.89	231	42.53	381	43.94	394
1	55.01	4647	60.03	3928	64.12	4217	65.10	4311	57.02	4198	58.91	4628	62.30	4481
op	40.13	1028	40.05	1707	40.10	1410	44.51	1370	39.03	1211	39.01	906	40.92	1001
1	55.38	4723	60.31	3965	64.54	5301	66.43	4428	57.15	4210	59.36	4713	62.48	4563
op	41.28	1243	40.17	1754	41.35	1507	45.30	1381	39.29	1316	39.87	987	41.05	1121

optimum WER is achieved by $k = 2$ and $k = 3$ models.

Info_tren is a more difficult task than Bdgeo. Moreover, the relationship between the number of parameters (k-grams) to be trained and the available training material is poorer for Info_tren than for Bdgeo. As a consequence, $k > 2$ models do not exceed the performance of $k = 2$ models in this database, because they are all poorly trained. However, major PP increases are found for high values of k for some of these models (Fig. 2).

Figures 9 and 10 show a subset of plots in Figs. 7 and 8 respectively. They only plot a trace per smoothing technique which corresponds to the best WER for the Bdgeo and Info_tren databases respectively.

Figures 9 and 10 show that models optimizing the test set perplexity (Figs. 1 and 2) also lead to the best system performance when the α parameter is optimized. However, correlations between PP and WER are not comparable to those reported in Klakow and Peters (2002). In these tasks, as in many real application tasks, the available training data do not allow us to get LM probability distributions close to the "true" distribution.

Figures 9 and 10 show similar WER for all smoothed LMs. Differences among the computational costs involved (AAN) are not very big but could be taken into account when the CSR system needs to work in small, low performance devices.

Kneser-Ney discounts lead to worse system performances, especially in the Bdgeo database (Fig. 10). In this database, they lead to higher WER for $k > 3$ models and to higher computational costs for $k > 2$ models. This behavior, predicted by the PP evaluation, seems to show that this discount is less competitive for high values of k. The number of different contexts in which a sequence of words appears increases quickly with k and therefore the distribution $\beta(\omega_i \mid h^*) = C(\omega_i \mid h^*)/C(h^*)$ in Eq. (12) could be worse estimated than distribution $P(\omega_i \mid h^*)$ in Eq. (7). Note that in our paper the distribution $\beta(\omega_i \mid h^*) = C(\omega_i \mid h^*)/C(h^*)$ is included in a back-off smoothing schema whereas in literature it is typically used in interpolated models. On the other hand only the absolute discount is implemented for the Kneser-Ney proposal.

Best results are summarized in Table 5 for both databases. However, differences around the optimum WER are not very significant in most cases.

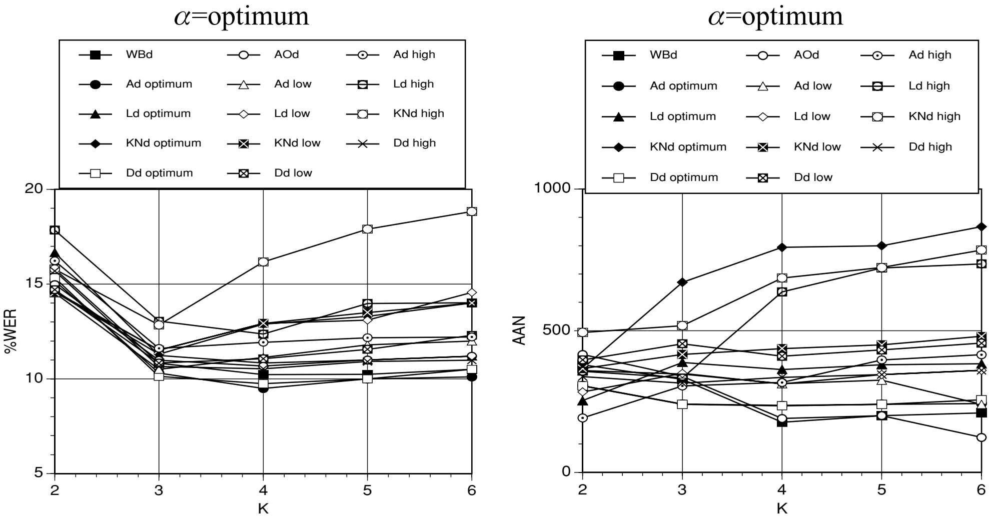

Figure 7. WER and AAN obtained for Bdgeo through the smoothed k-TSS LMs with $k = 2 \ldots 6$. The scaling factor is set to its optimum value in each case (emphasized values in Table 3). PP evaluation can be found in Fig. 1.

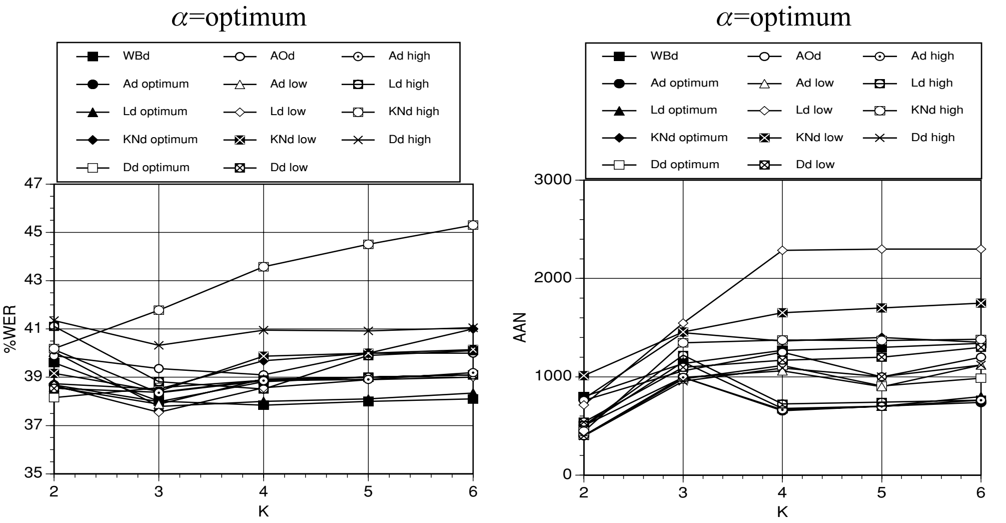

Figure 8. WER and AAN obtained for Info_tren through the smoothed k-TSS LMs with $k = 2 \ldots 6$. The scaling factor is set to its optimum value in each case (emphasized values in Table 4). PP evaluation can be found in Fig. 2.

6. Concluding Remarks

We have tried to analyse the effect of the smoothing technique applied to the Language Model in the CSR system and to show its real impact on final system error rates. The effect of the smoothing technique on system performance is not independent of subsequent scaling of LM probabilities. This relationship between the two effects has been analyzed in this work and their related contribution to final system performance has been established.

The back-off formalism is chosen because the recursive scheme involved is well integrated into the syntactic approach. Classical discounting-distribution schemes, as well as the recently proposed Delimited discount, have been compared in terms of both PP and final system performance measured through WER and the computational cost involved. To our knowledge this is the first time that the interdependency between smoothing technique, probability scaling at decoding time and final system performance has been analyzed.

Table 5. Best results obtained

	Bdgeo task			Info_tren task		
k	Smoothing	WER	ANN	Smoothing	WER	ANN
2	Low absolute $\alpha = 4$	14.54	359	Low absolute $\alpha = 4$	38.60	504
3	Optimized delimited $\alpha = 5$	10.13	240	Low linear $\alpha = 4$	37.55	1547
4	Optimized absolute $\alpha = 5$	9.49	236	Witten-Bell $\alpha = 5$	37.84	1303
5	Optimized absolute $\alpha = 5$	10.02	240	Witten-Bell $\alpha = 5$	38.01	1301
6	Optimized absolute $\alpha = 5$	10.10	246	Witten-Bell $\alpha = 5$	38.19	1340

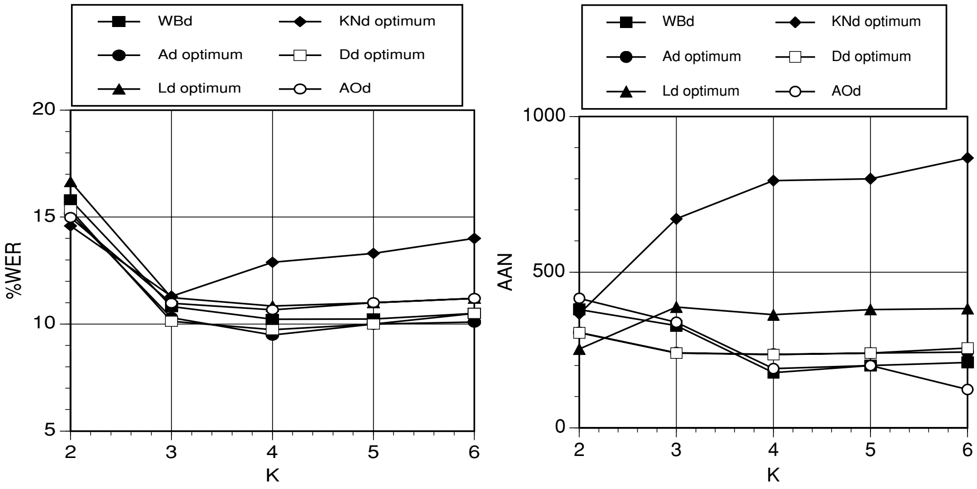

Figure 9. Selection of best WER for Bdgeo from Fig. 7: only a trace per smoothing technique. The PP evaluation of these models is shown on the right side of Fig. 1.

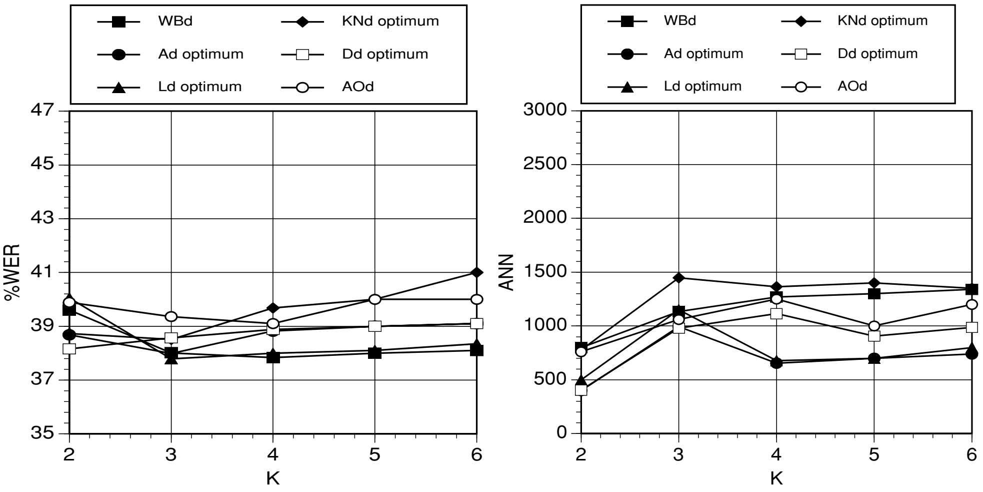

Figure 10. Selection of best WER for Info_tren from Fig. 8: only a trace per smoothing technique. The PP evaluation of these models is shown on the right side of Fig. 2.

Experiments show that the selection of the α parameter to get the lowest WER is also dependent on the smoothing technique and is not independent of the task and available training data.

We have found a strong dependence between the smoothing technique and the value of the scaling parameter α needed to get the best system performance, which is, indeed, perplexity independent in many cases. On the other hand, experiments have shown that large, significant differences among PP values could lead to small differences in final system WER, i.e., "bad LMs" could also lead to quite good final system performances.

The inter-dependences established in this work should be considered when optimizing and adjusting CSR systems to work on real tasks. In such cases, the task and the available training data lead to LM probability distributions far removed from the "real" distribution and thus PP and WER may not correlate very well. Moreover, the computational cost involved should be considered especially when real applications also require the integration of the CSR system into a small, low performance device.

Acknowledgments

This work has been partially supported by the Spanish CICYT under grant TIC2002-04103-C03-02 and TIN2005-08660-C04-03.

Notes

1. This is also related to the problem of the negligible impact that transition probabilities have in acoustic models.
2. This technique is applied in Katz's discounting when all events appear after a history h more than r times (Ney et al., 1997).
3. Good-Turing discount: $(1-\lambda) = [N(\omega_i|h)+1]\frac{n_{N(\omega_i|h)+1}}{n_{N(\omega_i|h)}}$ where n_i is the number of events that occur exactly i times in the training data. The final mass of probability assigned to *unseen* events is equal to $n_1/N(h)$, where n_1 is the number of events seen once and $N(h)$ is the total number of events after history h.

References

Bimbot, F., El-Béze, M., Igounet, S., Jardino, M., Smaili, K., and Zitouni, I. (2001). An alternative schme for perplexity estimation and its assessment for the evaluation of language models. *Computer, Speech and Language*, *15*(1):1–13.

Bonafonte, A., Aibar, P., Castell, N., Lleida, E., Mariño, J., Sanchis, E., and Torres, I. (2000). Desarrollo de un sistema de diálogo oral en dominios restringidos. In *Primeras jornadas en Tecnologı́a del Habla*.

Bordel, G., Torres, I., and Vidal, E. (1994). Back-off smoothing in a syntactic approach to Language Modelling. In *Proc. International Conference on Speech and Language Processing,* pp. 851–854.

Chandhuri, R. and Booth, T. (1986). Approximating Grammar Probabilities: Solution of a Conjecture. *Journal ACM*, *33*(4):702–705.

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

Chen, F. S., and Rosenfeld, R. (2000). A survey of smoothing for me models. *IEEE Transactions on Speech and Audio Processing*, *8*(1):37–50.

Clarkson, P., and Robinson, R. (1999). Improved language modelling through better language model evaluation measures. In *Proc. of European Conference on Speech Technology,* Vol. *5*, pp. 1927–1930.

Clarkson, P., and Rosenfeld, R. (1997). Statistical language modeling using the CMU-Cambridge toolkit. In *Proc. of European Conference on Speech Technology,* pp. 2707–2710.

Díaz, J., Rubio, A., Peinado, A., Segarra, E., Prieto, N., and Casacuberta, F. (1998). Albayzin: A task-oriented spanish speech corpus. In *First Int. Conf. on Language Resources and Evaluation,* vol. II, pp. 497–501.

Dupont, P., and Miclet, L. (1998). Inférence grammaticale régulière. fondements théoriques et principaux algorithmes. Rapport de reserche N 3449, Institut National de recherche en informatique et en automatique, Rennes.

García, P., and Vidal, E. (1990). Inference of *k*-testable languages in the strict sense and application to syntactic pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(9):920–925.

Goodman, J. (2001). A bit of progress in language modeling. *Computer, Speech and Language, 15*:403–434.

Hopcroft, J., and Ullman, J. (1979). *Introduction to Automata Theory, Languages and Computation*. Addison-Wesley.

Jelinek, F. (1985). Markov source modelling of text generation. In J.K., Skwirzynski, and M., Nijhoff, (Eds.), *The Impact of Processing Techniques on Communication,* Dordrecht, The Netherlands, pp. 569–598.

Jelinek, F. (1996). Five speculations (and a divertimento) on the themes of h. bourlard, h. hermansky and n. morgan. Speech Communication *18*:242–246.

Jelinek, F. and Mercer, R.L. (1980). Interpolated estimation of markov source parameters from sparse data. In Workshop on Pattern Recognition in practise. The Netherlands: North-Holland, pp. 381–397.

Jurafsky, D. and Martin, J.H. (2000). *Speech and Language processing: An Introduction to Natural Language Processing, Computational Linguistics and Speech Recognition.* New Jersey:Prentice Hall.

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

Klakow, D., and Peters, J. (2002). Testing the correlation of word error rate and perplexity. *Speech Communication, 38*:19–28.

Kneser, R., and Ney, H. (1995). Improved backing-off for m-gram language moleling. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*. Detroit, M.I., vol. I., pp. 1783–1786.

L. Mangu, E. B., Stolcke, A. (2000). Finding consensus in speech recognition: Word error minimization and other applications bof confusion networks. *Computer, Speech and Language, 14*:373–400.

Ney, H., Martin, S., and Wessel, F. (1997). Statistical Language Modeling using leaving-one-out. In S., Young, G., Bloothooft, (Eds.), *Corpus-based Methods in Language and Speech Processing*. Kluwer Academic Publishers, pp. 174–207.

Ogawa, A., Takeda, K., and Itakura, F. (1998). Balancing acoustic and linguistic probabilities. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. II, pp. 181–185.

Rodríguez, L., Torres, I., and Varona, A. (2001a). Annotation and analysis of disfluencies in a spontaneous speech corpus in spanish. In: Disfluency in Spontaneous Speech. An ISCA Tutorial and research workshop.

Rodríguez, L., Torres, I., and Varona, A. (2001b). Evaluation of sublexical and lexical models of acoustic disfluencies for spontaneous speech recognition in spanish. In *Proc. of European Conference on Speech Technology,* vol. 3, pp. 1665–1668.

Rodríguez, L., Varona, A., de Ipiña, K.L., and Torres, I. (2000). Tornasol: An integrated continuous speech recognition system. In Pattern recognition and Applicactions. Frontiers in Artificial Intelligence and Applications series. The Netherlands:IOS Press, pp. 271–278.

Rosenfeld, R. (2000). Two decades of statistical language modeling: Where do we go from here. *Proceedings of the IEEE*, *88*(8).

Rubio, J.A., Diaz-Verdejo, J.E., GarcÌa, P., and Segura, J.C. (1997). On the influence of of frame-asynchronous grammar scoring in a csr system. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing,* vol. II, pp. 895–899.

Torres, I. and Varona, A. (2000). An efficient representation of k-TSS language models. Computación y Sistemas (Revista Iberoamericana de Computación), *3*(4):237–244.

Torres, I., and Varona, A. (2001). k-tss language models in a speech recognition systems. *Computer, Speech and Language, 15*(2):127–149.

Varona, A., and Torres, I. (1999). Using Smoothed k-TLSS Language Models in Continous Speech Recognition. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing,* vol. II, pp. 729–732.

Varona, A., and Torres, I. (2000). Delimited smoothing technique over pruned and not pruned syntactic language models: Perplexity and wer. In Automatic Speech Recognition: Challenges for the new Millenium, ISCA ITRW ASR2000 Workshop, Paris, pp. 69–76.

Varona, A., and Torres, I. (2001). Back-off smoothing evaluation over syntactic language models. In *Proc. of European Conference on Speech Technology*. Vol. 3, pp. 2135–2138.

Varona, A., and Torres, I. (2003). Integrating high and low smoothed lms in a csr system. In *Progress in Pattern Recognition, Speech and Image Analysis. Lecture Notes in Computer Science*, vol. 1, pp. 236–243.