Directory Name Retrieval over the telephone in the Picasso project

F. Neubert, G. Gravier, F. Yvon, G. Chollet

Ecole Nationale Supérieure des Télécommunications / CNRS – URA 820
46, rue Barrault
75634 Paris Cedex 13, France
chollet@sig.enst.fr

ABSTRACT

The European project PICASSO intends to develop and test several telematics transaction services that will be accessible via the worldwide telephone network.
In this framework, ENST works on developing an Automated Speech Recognition system of pronounced and spelled names, for telephone quality speech in French. The recognizer is based on Hidden Markov modeling of speech units using word models for spelled letters and phone models for name pronunciation. Bigram probabilities are introduced at this stage for phonemes and letters, in order to improve the quality of decoding.
The directory was built automatically from the list of the names contained in the database, using a grapheme to phoneme converter for the names and rules for spellings, each entry in the directory consisting of several pronunciations and spelling variants.
After the acoustic recognition phase, the corresponding entry in the directory is then found using dynamic alignment of symbol sequences, with insertion, deletion and substitution costs determined from the training data to take into account acoustic confusability.
As this lexical search is very time consuming for large directories, we present a faster method using pre-selection in a tree-based representation of the lexicon. A rescoring strategy on the 10 best outputs is also evaluated.

I. INTRODUCTION

Automatic retrieval of names from their pronunciation and spelling is a difficult problem in speech recognition, possible applications being name retrieval through the telephone network for purchase, financial, or multi-media telecommunication services.
Such systems need to integrate various speech technologies (e.g. speech recognition, speaker verification, text-to-speech synthesis) into intuitive, user friendly and secure interfaces.

Many studies address the general problem of alphabet recognition, which is known to be a difficult task because of acoustic similarities between some letters (e.g. the well known E-set in English). When working with telephone quality speech, the confusability increases drastically.
Cole *et al.* presented experiments with telephone speech for recognition on French and English alphabets [1,2]. Letters, separated by pauses, are segmented and then classified using a MLP with phonetic features. More recently, HMM based methods have been proposed, as in [3] and [4], where letter models are used. The first of these studies compares DTW- and HMM-based lexical search strategies to retrieve names from natural spelling. HMM-based lexical search can also be seen as a DTW whose insertion, deletion and substitution costs are learnt from the training data. The second study is based on a multilevel classification with a N-best approach, using DTW and a restricted grammar for lexical search.
These studies however fail to address a number of problems that are critical in the perspective of real applications. First, they greatly underestimate the variability observed in real-life spellings. Second, phonetic knowledge is only used to help letter discrimination but not for the DTW-based lexical access. Finally, the name itself (i.e. the pronunciation of the name) is a source of information that has rarely been used (see however [5]).
In this paper, experiments on name retrieval from the pronunciations and the spellings are reported, using Hidden Markov Models of speech units and robust lexical access based on phoneme or letter confusability. The name directory is build automatically and integrates pronunciation and spelling variants.
The paper is organized as follows. In the next section, the database is described and the construction of the directory is explained. Acoustic modeling is then presented, along with some results. In section 4, different retrieval strategies are evaluated. Experiments on rescoring are then discussed, and their results commented. Finally, other possible improvements of this baseline system are mentioned.

II. DATABASE AND DIRECTORY

The Swiss-French Polyphone database[1] was recorded over telephone lines with 5,000 speakers calling once, a call including the pronunciation and spelling of 3 names, where a name can be a person name, a city name or any common noun. Spellings were prompted but no specific guidelines were given to the callers. As a result, in addition to "standard" spellings, the corpus contains occurrences of comparisons, such as "*a comme Alain*" (a not so uncommon way to minimize confusions between letters), occurrences of aeronautic-like spellings, e.g. "*alpha bravo…*" *etc.*

A subset of the database is used, containing 11,920 speech segments from 3,998 speakers. This subset was divided in three corpora. The first one is the train corpus, containing 5,390 segments from 3,223 speakers. The remaining items belong to the test corpus, from which a special subset, the "*clean*" test corpus, is extracted. This "*clean*" test corpus contains the items for which the spelling conforms to the "standard" spelling conventions. It contains 5,015 segments from 3,097 speakers, corresponding to 3,478 different names and is used to evaluate the quality of acoustic modeling.

The directory from which names are to be retrieved contains the 8,261 entries presented to the callers. Given the typical size of directories and the fact that only few occurrences of each name are available for training, we cannot rely upon whole word based recognition. Therefore, each directory entry contains one or several phonetic transcription(s) of the name, plus one or several possible spellings. These phonetic transcriptions were automatically produced by a grapheme to phoneme converter developed during the course of the Onomastica project [6,7]. This transcriber has been explicitly devised to cope with proper names idiosyncrasies. It also contains a module for recognition of proper names origins and is likely to output pronunciation variants. The spelling variants are generated by a rule-based system. For each letter (or cluster of letters), all possible spellings are considered. For instance, the letter "é" can be spelled "*e accent aigu*", "*e aigu*" or "*é*"; the cluster "nn" can be spelled "*deux n*" or "*n n*"… In the remainder of the paper, a directory entry E_i will be referred to as $\{N_{i,j=1,\ldots,ni}, S_{i,j=1,\ldots,si}\}$, where $N_{i,j}$ (resp. $S_{i,j}$) is the j-th pronunciation (resp. spelling) variant.

III. ACOUSTIC MODELING

3.1 Models and grammar

The first processing stage consists in acoustic decoding. Speech is encoded using 12 Mel frequency cepstral coefficients and log energy, with first and second order derivatives, computed every 10 ms on 25.6 ms frames. In order to estimate phone models for recognition of names and letter models for recognition of spellings, the training corpus is first automatically segmented from its orthographic transcription, using speaker-independent HMMs of phones. The latter are also used as bootstrap models. Phone model parameters are then re-estimated using the standard Baum-Welch algorithm. Letter models are created by concatenating the bootstrap models that correspond to the letter pronunciation, and performing re-estimation afterwards. Because of the possible liaisons after some words such as "*deux*", some letters can have two models, one with liaison and one without. A simple grammar is also used during the decoding procedure. This grammar describes an utterance as a name followed by a spelling where a name can be any sequence of phonemes and a spelling any sequence of letters. An optional silence is also specified between the name and the spelling, which has been shown to improve the recognizer's detection of spelling's start [8].

It is important to realize that our grammatical model allows one or the other part of the utterance to be skipped by the speaker, a very likely situation in real applications. A short silence is forced after each letter in the spelling. In fact, a preliminary study inspired from [9] showed that forcing a short silence gives better accuracy for letter recognition. Using the grammar defined above, the recognizer finds out the most likely sequence in the speech signal.

3.2 Results and discussion

In this section, results concerning speech unit recognition are presented and discussed, allowing for an independent evaluation of the acoustic models. Tables 1 and 2 report various phone and letter recognition rates and accuracies on the train and test corpora.

The accuracy corresponds to the number of correctly recognized units minus the number of insertions and is a much more reliable measure of the recognition quality than the correct recognition rate. As can be seen on these tables, although the correct recognition rate is quite good, there are many phoneme insertions.

Reducing the number of insertions can be done by introducing a penalty for the transition between two models of the form $p+s*\log(P[W_i|W_{i-1}])$, where p is a fixed penalty, $P[W_i|W_{i-1}]$ the bigram probability of speech unit W_i following W_{i-1}, and s a scale factor which determines the importance given to the bigrams.

	phonemes		letters	
Penalty factors:	correct.	accuracy	correct.	accuracy
p=0 s=0	58.71	14.42	83.16	74.56
p=-17.5 s=0	54.35	41.50	83.44	80.27
p=0 s=8	59.31	51.52	89.74	84.09

Table 1- Recognition rate (in %) and accuracy for phonemes and letters, on the train corpus.

[1] The Swiss-French Polyphone database, recorded by Swiss Telecom PTT, is distributed by the European Language Resource Association. *http://www.icp.inpg.fr/ELRA*

Separate optimizations on p (see [10]) and s (see below, figure 1) yielded the results mentioned above. Experiments combining a fixed penalty with bigrams (p and s non-null) showed lower values of accuracy than the best bigrams approach without fixed penalty (p=0 and s=8). This can be explained by the fact that p is a kind of fixed bigram probability, the best strategy being then to take into account the real transition probabilities between speech units, which were computed on a lexicon of 50,000 words. For this task, we used the CMU Statistical Language Modeling Toolkit along with a linear discounting method, which gave the lowest perplexity for the resulting language model [11]. Finally, figure 1 plots the correct recognition rates and accuracies on the train corpus, as a function of the scale factor s (with p=0). The corresponding name retrieval rates are also plotted. These results point out the facts that letter recognition is much more reliable than phone recognition, and that the language model helps to improve greatly the accuracy of phone recognition.

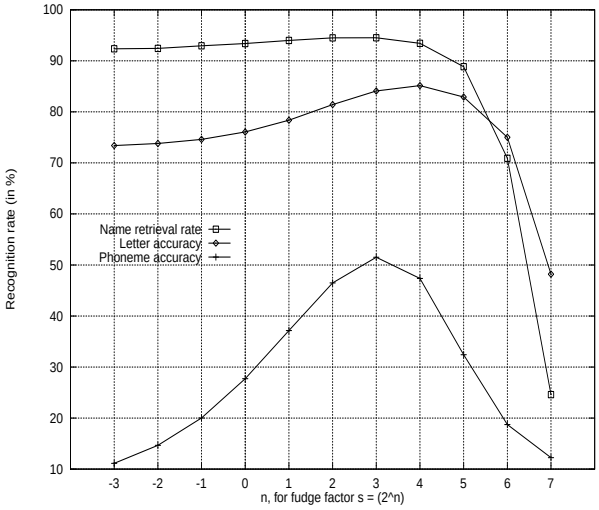

Fig. 1 - Recognition rates on the train corpus as a function of the bigram scale factor.

Hereafter the recognition rates for speech units on the clean test corpus are presented (Table 2). As specified above, these results directly depend on the quality of acoustic decoding.

Penalty factors	Phonemes		Letters	
	correct.	accuracy	correct.	accuracy
p=-17.5 s=0	56.24	34.39	82.77	78.43
p=0 s=8	60.21	47.55	88.01	81.90

Table 2 - Recognition rate (in %) and accuracy for phonemes and letters, on the clean test corpus.

Using a fixed or a bigram based transition penalty does not affect much letter recognition rates. For phonemes however, the accuracy is much better when bigrams are used, but the difference between phones and letters is still of 30%. These remarks are important, as they will help us comment the name retrieval rates we obtain when using these different decoding procedures (see section IV). We must now deal with another aspect of acoustic decoding quality, which concerns the detection of spelling start. Indeed, the number of phone insertion can be explained by the fact that the recognizer often detects the beginning of the spelling too late. This is a consequence of the fact that phones may be recognized where letters should be recognized…For that reason, phonemes tend to be overestimated, while letters in the spelling can be lost when confused with phoneme sequences. The last improvement concerning that problem was the specification of a silence in the grammar between name and spelling, which decreased the error rate due to an erroneous identification of spelling start from 65% of the errors, down to about 35% [8]. Still, this problem remains quite important, as shown below.

In order to estimate the importance of a correct discrimination between name and spelling during recognition, a simulation was made, which consisted in splitting artificially the encoded signal corresponding to speech in two parts: one, corresponding to the name' s pronunciation, and the second one, corresponding to its spelling. This artificial splitting was done using the label files obtained from forced recognition. These label files were used when training the models. As the forced recognition was based on transcriptions of what was prompted, we used here an *a posteriori* knowledge concerning how the speakers behaved. Table 3 reports the following results.

Penalty factors p=0 s=8	Phonemes		Letters	
	correct	accuracy	correct	accuracy
actual splitting	60.21	47.55	88.01	81.90
artificial splitting	63.06	50.21	88.22	83.38

Table 3 - Recognition rate (in %) and accuracy with natural decoding and artificial splitting.

It appears that the bigrams strategy tends to improve the natural discrimination between phonemes and letters, as there is just a slight difference between the accuracies obtained through a normal decoding, or with the procedure using an artificial splitting. A comparison made on lexical access, reported in next section will conclude on this quite important question.

A more detailed study of the letter recognition errors outlines the standard confusions between acoustically similar letters such as "b" and "d", "f" and "s" or "p" and "t". As noted in many studies on alphabet recognition (e.g. [3, 12]) letter HMMs are not really able to distinguish two letters whose spelling only differ by a short transitional acoustic event. For example, letters "m" and "n" share the same vowel and are separable by the last consonant. But phonemes /*m*/ and /*n*/ are quite similar, and their confusion is also one of the most common substitution at the phoneme level.

IV. LEXICAL SEARCH

4.1 Search strategy

The second step consists in retrieving the name in the directory from the recognized strings. This is done by calculating the dissimilarity measure between the recognized form *R* and each lexical entry *Ei* according to:

$$D(R,Ei) = \beta \min_j d(Rs,Si,j) + (1-\beta) \min_j d(Rn,Ni,j)$$

where Rs (resp. Rn) is the recognized spelling (resp. pronunciation). The retrieved name is the one for which the dissimilarity is the smallest. Parameter β allows to balance the respective contributions of the spelling and pronunciation. Dissimilarity d(.,.) is computed by dynamic alignment using specific costs for each possible substitution, insertion and deletion. Those costs are usually arbitrarily fixed. However, as some pairs of symbols (letters or phonemes) are more confusable than others, it seems natural to assign a smaller cost for the substitution of confusable symbols.

Therefore, the weighted cost for the substitution of *x* by *y* is -log (p(*y*|*x*)), where p(*y*|*x*) is estimated on the train corpus. The same procedure is used to determine insertion and deletion costs. This approach is somewhat similar to the HMM-based alignment procedure in [3].

Instead of using phonetic knowledge to achieve better discrimination as in [12], this knowledge is learnt from the training data and used for name retrieval rather than for symbol recognition. Setting β = 0.5 and with no transition probabilities between models (p=s=0), name recognition rate achieves 52.11% with binary confusion costs, and 82.11% with weighted costs.

Figure 2 gives the name retrieval rate as a function of β, with p= -17.5 and s=0 on the train corpus. The best results are achieved for β=0.6, with 94.17% of names retrieved on the train corpus, and 82.68% on the test corpus.

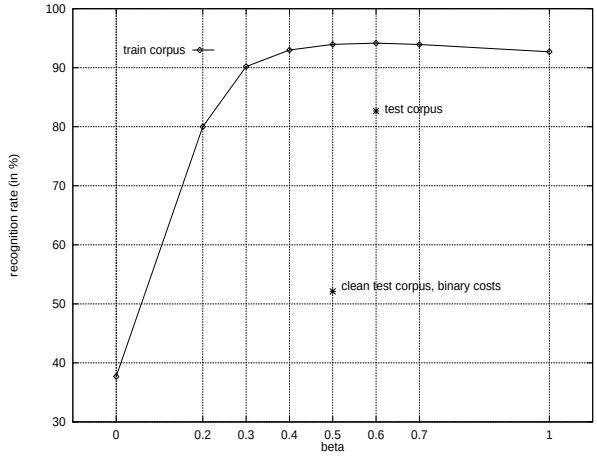

Fig 2 – Name recognition rate as a function of β.

Using just the pronunciation (β=0), the rate is less than 40%, while a spelling only recognition (β=1) goes beyond 90% of retrieved names. This is due to a more accurate letter recognition that we noticed in section *3.2*. The spelling can therefore be seen as the main source of information, while the pronunciation brings some kind of redundancy, which improves slightly the global retrieving process.

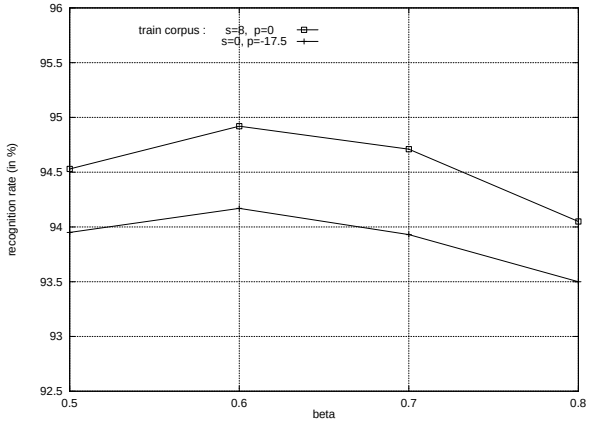

Fig 3 - Name recognition rates as a function of β, for best decoding using fixed or bigrams-dependant transition penalties.

The integration of the bigram language model described in the previous section gave the following results concerning lexical access: the best rate for a decoding using bigrams is also obtained for β=0.6 (see figure 3). On the test corpus, the corresponding rate is now 83.53%, which shows an increase of less than one point. This result can easily be explained if one remembers (see *3.2*) that introducing our bigram language model really improved the phone accuracy, while the corresponding value on letters was just slightly modified, but was still much higher than the performance on phonemes. That is why here the global retrieval rate shows only a small improvement, since letters remain the main source of information for lexical search.

Anyway, as it is shown below, any improvement of speech units decoding accuracy should not be neglected, even at that stage where the global performances of the system are not much modified. Indeed, the experiment described in section *3.2*, which consisted in performing an artificially exact splitting between name and spelling during the decoding procedure gave a low increase, on both phonemes and letters accuracy. When performing lexical access on this basis, with the previously optimized parameters concerning the bigrams strategy (p=0, s=8, β=0.6), we obtain a retrieval rate of 90.43% on the clean test corpus, while we have 89.11% with a natural splitting on the same corpus.

This result shows that the integration of a language model both improved the accuracy on speech units and the detection of spelling start, as we' ve seen here that an artificially correct splitting brought low increase both on the speech units accuracy and on the names retrieval rate.

4.2 Pre-selection

The name retrieval procedure described in the previous section is extremely time consuming, as it requires the computation of dynamic alignments between the recognized spelling and pronunciation, and each variant of each lexical entry. Obviously, this brute force approach is only applicable in the context of small or medium sized name directories. As a first step towards very large directory name retrieval, an alternative search strategy is evaluated in this section. This new strategy consists in the two following stages: first, on the basis of the spelling alone, a set of *N* possible entries are selected using a fast approximative lexical search; this subset is then exhaustively explored, according to the procedure defined in the previous section. The approximative lexical search implements a variant of the algorithm originally proposed in [13] for error tolerant lexical recognition. Given a possibly erroneous form, this algorithm efficiently retrieves every lexical entry whose edit distance (computed as the number of deletions, insertions and substitutions between two forms) to the form falls below a predefined threshold d.

This search procedure crucially relies upon a finite state representation (tree-based, in this implementation) of the lexicon, which makes it possible to optimize the computation of edit distances. Furthermore, the use of uniform unitary costs for weighting insertions, deletions and substitutions allows to compute, given the edit distance between *x* and a lexical prefix *s*, a lower bound of the minimum edit distance between the searched form and every lexical entry whose prefix is *s*. As a consequence, the traversal of the tree can be dramatically speeded up, since the exploration of entire branches can be stopped after the computation of the distance between the form and lexical prefixes.

The pre-selection implementation follows the same lines, with one slight difference: we are interested in retrieving a fixed number *N* of lexical neighbors, rather than all the neighbors within a given distance. Accordingly, the threshold *d* is initially set to be equal to the *N*-th closest distance, and let this threshold decrease as better lexical candidates are retrieved. If this patch appears to somehow slow down the search, the pre-selection stage remains however extremely fast. Retrieval of an entry is about 9 times faster with the pre-selection of 100 lexical neighbors with a very small decrease in terms of recognition rate.

Figure 4 plots the recognition rates as a function of the number of neighbors selected. As *N* increases, the recognition rate tends toward the rate obtained with no pre-selection. With only 250 neighbors, the recognition rate is very close to the optimal one and the computation times are severely reduced.

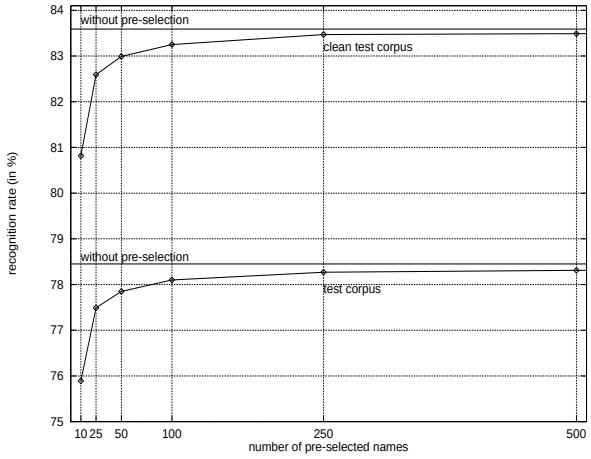

Fig 4 – Retrieval rates as a function of the number of pre-selected candidates.

V. RESCORING STRATEGY

In order to evaluate a possible improvement of the baseline system described above, a rescoring on the 10 closest names outputted by the previously described lexical search was finally performed [4]. This strategy consists simply in a second decoding pass, based on constraint grammars: for each item in the test corpus, a specific grammar is used for its decoding, which forces the recognizer to choose one entry among the ten closest entries given as output of the first pass' s lexical search. The best entry among the 10 possible ones is chosen by calculating the recognition scores over the selected HMMs when the recognizer performs Viterbi alignment. This experiment was designed as a second pass following the one described in the previously discussed recognitions, with p=-17.5, s=0, and β=0.6.

Performing this experiment gave us the following results: the lexical access rate showed a value of 82.09% on the corpus used (as rescoring is very time consuming, we used a large subset of the test corpus). Knowing that the rate for the first pass's lexical search was 82.75%, with a confidence interval that is [81.76%, 83.59%], we can say that rescoring brought no significant change on lexical access performance.

The rank correlation matrix shows that the first output remains in first position for 82% of the rescored items, which is normal as it may correspond more or less to the 82% of correct outputs given by the first pass. It appears, therefore, that rescoring fails when the decoder tries to re-estimate the scores on the remaining outputs in order to find the best one.

This can be interpreted as the sign of a limit reached in the quality of decoding, related to the quality of the models themselves. That is, performing rescoring seems to be useless, since the same models are used in both passes and that they are likely to make the same errors.

VI. CONCLUSION

Experiments on letter recognition using Hidden Markov Models outlined that this technique does not avoid standard confusions between phonetically similar letters. However, the knowledge of letter confusability can be used to improve the lexical search by adapting the dynamic alignment costs to the possible confusions. A decoding taking into account bigram probabilities of phonetic units gave good results for speech units accuracy (especially for phonemes) and also for spelling detection. In addition, an algorithm for the selection of a list of possible entries speeds up the computation time required for a lexical access with a very small loss in the recognition rate. A rescoring experiment suggested also that a limit had been reached in the quality of the models used here in order to improve the decoding procedure. The evaluations presented in this study were made with real-life (i.e. unconstrained) natural pronunciations and spellings. In this framework, something should be done in order to deal with non-standard spellings. With the grammar defined, such spellings may be recognized as a sequence of phones or letters, thus introducing errors in the lexical search. It seems obvious that a system able to detect and ignore (or use) non-standard spelling should perform better on the entire test corpus.
Such an Automated Speech Recognition system, designed for real-life conditions, could be integrated as a front end to a speaker verification system, in order to hypothesize an identity, as needed in the PICASSO project.

ACKNOWLEDGEMENT

This work is partially supported by SwissCom.

REFERENCES

[1] R. Cole, K. Roginski, and M. Fanty, "English alphabet recognition with telephone speech", in Eurospeech, 1991, pp. 479-482.

[2] P. Schmid et al., "Real-time, neural network-based, French alphabet recognition with telephone speech", in Eurospeech, 1993, pp. 1723-1726.

[3] D.Jouvet et al., "Speaker independent spelling recognition over the telephone", in Int. Conf. on ASSP, vol. 2, 1993, pp.235-238.

[4] J. C. Junqua et al., "An n-best strategy, dynamic grammars and selectively trained neural networks for real-time recognition of continuously spelled names over the telephone", in Int. Conf. on ASSP, 1995, pp.852-855.

[5] M. Meyer and H. Hild, "Recognition of spoken and spelled proper names", in Eurospeech, 1997, pp. 1579-1582.

[6] M. Schmidt, S. Fitt, C. Scott, and M. Jack, "Phonetic transcription standards for European names (ONOMASTICA)", in Eurospeech, vol. 1, 1993, pp. 279-282.

[7] F. Yvon, *Prononcer par analogie: motivation, formalisation et évaluation*, PhD thesis, ENST, 1996.

[8] F. Neubert, G. Gravier, F. Yvon, and G. Chollet, "Reconnaissance de noms prononcés et épelés au téléphone par modèles de Markov cachés", in JEP, 1998.

[9] D. Pye, *Automatic recognition of continuous spelled Swiss-German letters*, Technical report, IDIAP, 1994.

[10] G. Gravier et al., "Directory Name Retrieval using HMM modeling and robust lexical access", in Workshop on ASRU, 1997, pp. 558-565.

[11] H. Ney, U. Essen, and R. Kneser, "On structuring probabilistic dependencies in stochastic language modeling", in Computer Speech and Language, vol. 8, 1994, pp. 1-28.

[12] P. C. Loizou and A. S. Spanias, "High-performance alphabet recognition", IEEE Trans. on Speech and Audio Processing, 1996, pp. 430-445.

[13] K. Oflazer, "Error-tolerant finite-state recognition with applications to morphological analysis and spelling correction", *Computational Linguistics*, 1996, pp. 73-89.