A New View of the Output from Word Recognition

Marc-Peter Schambach
Siemens AG, L&A, Postal Automation
78459 Konstanz, Germany
Marc-Peter.Schambach@siemens.com

Abstract

Word recognition has changed in recent years. Implementations have become better, which allows larger vocabularies to be recognized; many of the implementations now available are suited well to specific tasks, but several have to be combined to obtain maximum benefit; and they are still hampered by deficiencies, for example when trying to decide among similar words. The new view of the output interface from word recognition presented here aims at various goals: easier combination of comparable word classification systems, better post-processing of word recognition results at higher levels (where more context is available), and improvement of out-of-vocabulary behavior. The proposed solution is to provide two different result scores instead of one single value: an overall quality measure, indicating the credibility of recognition, and a similarity measure for each word alternative. This shifts the responsibility for decisions from low-level word recognition tasks to higher levels, while retaining all necessary recognition information. This, however, gives rise to the need to define new performance metrics for evaluation. Implementations of the proposed output interface for two recognition engines are sketched.

1. Introduction

In the area of handwritten word recognition, much effort has been spent on the development of classification methods and algorithms. More and better algorithms have been developed and are widely available for use. As a consequence, the range of possible applications has expanded, and the demands on the word recognition task have grown, which has led to a change in the perceived requirements. This paper demonstrates the need for a different view of word recognition, from the perspective of its clients. The question it addresses is simple: What is the optimal client-side interface to word recognition? A proposal for such an interface will be made.

This paper is organized as follows: In section 2, the traditional view of word recognition is described, and its shortcomings are analyzed in section 3. This leads to a definition of the goals for this work in section 4. Section 5 presents a proposal for a new definition of recognition scores. This makes it necessary to define new performance metrics for these scores in section 6. In section 7, possible implementations of the new scores for selected word recognition systems are presented.

2. Word recognition

In this section, traditional views of word recognition will be described: What methods and implementations exist currently, what inputs do they handle, what results do they provide, and how can their performance be defined.

Traditional implementations of recognition engines can be divided into two categories:

- Two-step approaches: First, individual characters are recognized, and these results are then correlated to words and complete interpretations.
- Integrated approaches, which recognize whole words in one step: These approaches include holistic methods, which store representations for individual words, and methods like HMMs, which can recognize arbitrary patterns.

The input to word recognition has changed from small vocabularies, typically specified in lists of possible target words, to abstract lexical patterns, formulated for example as regular expressions. Word recognizers differ in the range of lexical patterns they can handle, in respect to restrictions on the alphabet and on the capabilities of the regular expressions they can implement. This blurs the distinction between unconditional (‘nominal’) recognition as compared to recognition with restricted reference patterns. Thus, this distinction will be regarded here simply as one between different performance modes of recognizers.

The output of a word recognizer consists typically of several result alternatives, each paired with a confidence value representing the estimated posterior probability for correctness. Much work on confidence values exists, e.g. [3, 4, 6].

In some systems, the posterior probability is calculated directly, e.g. with neural nets [8], while in others confidence values are derived from features, e.g. of a HMM recognition process [2]. To be able to interpret such values as probabilities, confidence mapping can be performed.

Recognition engines differ in recognition power. Their performance is typically characterized by parameters like recognition rate and error rate, which imply a definition of correctness. Depending on the particular task, the engines are optimized for different conditions. Some are optimized for reference verification, some for nominal recognition, others for their out-of-vocabulary (OOV) behavior.

3. Motivation

To better grasp the shortcomings of traditional views of recognition, we need to look at how they have defined 'word recognition results'. This definition has repercussions for the following topics:

3.1. Comparability of results

With several word recognition engines available – as is often the case nowadays – a problem arises when different word recognition engines with different capabilities have to be combined. How can their results be interpreted unambiguously, such that they can be compared to each other?

3.2. Distinction of recognition tasks

The variety of applications has led to varying requirements for word recognizers, but they are used basically in just two contexts:

- The word recognition system has to decide which word from a given reference description (typically a list of valid words) gives the correct match. Each result word is given a probability for correctness, called 'confidence' [2]. This mode will be called *reference recognition*, because much information is gained from the reference pattern.
- Other systems work without reference descriptions and are seen as performing a 'lexicon free' kind of recognition. Within these systems, only a very rough lexical description is given, specifying merely the allowed alphabet of characters (e.g. only numerals) or more complicated patterns like those used in language models based on transition probabilities (e.g. n-grams [1]). This mode will be called *nominal recognition*, because only information from the input image 'as is' is presented at the output.

These different modes present different interfaces. Their results are not comparable, and they have to be evaluated differently. But the distinction between them is fluid: Reference recognition with huge lists of valid words becomes similar to nominal recognition, while nominal recognition with a very restrictive lexical description becomes a form of reference recognition. Sometimes implementations mix the two methods for efficiency reasons: For the recognition of German postcodes (consisting of 5 digits, with about 43000 valid codes), a nominal recognition pattern '5 digits' can be followed by a reference check against the list of valid codes.

3.3. Effect of redundancy

One could assume that the size of the language generated by a reference pattern is the characterizing value for recognition type. So if a language consists of only two words, e.g. the German city names 'Hamburg' and 'Frankfurt', this could be called reference recognition, because the two strings have no redundancy. But if these cities were 'Hamburg' and 'Homburg', it would be correct to call it nominal recognition, because except in the second letter the strings are redundant, i.e. the recognizer does not need to refer to the reference pattern to accept the characters.

Thus, a characterizing value for the recognition mode is the amount of redundancy in the reference pattern. Properly seen, this is not a matter for the word recognizer, however, but for its clients. Independence from the reference pattern is therefore an important requirement for word recognition and will determine the further course of this study.

3.4. Dependence on reference patterns

Making the recognition results dependent on given reference patterns raises another problem. Often there is the need to use feedback from upper-level contexts, so that – depending on context – different reference patterns are specified for the same input image. Constantly changing result values ultimately create confusion, however, and caching of results becomes impossible. This is another reason for wanting to make recognition results independent of reference patterns.

3.5. Reject behavior

In word recognition, reject (OOV) situations arise when the word images do not match any reference pattern. These situations can be further divided into two categories:

- The writing style cannot be identified. This happens, for example, when there are unusual distortions in the writing, for which the recognition engine has not been trained, or when the word image doesn't represent a word at all.
- The written content cannot be matched. This is the common meaning of OOV, when the content of the word image is evident for the human reader, but doesn't match any reference pattern.

Thus a poor match to the reference pattern can have at least two causes: bad writing or a genuinely out-of-vocabulary word item. It is difficult to define all typical OOV environments (one would have to provide a representative set of all possible failures) and optimize the recognition engine to it.

3.6. Interpretation of result quality

Sometimes the client wants to know why a word hasn't been recognized reliably. Was poor image quality responsible, or was it due to the existence of similar interpretations? It seems that a single confidence value is not enough to report the quality of recognition results.

4. Goals and Requirements

Motivated by the considerations above, the topic of this work is the definition of a generalized output interface to report the results of word recognition. Several goals are targeted:

- Simplify the combination of word classifiers.
- Improve the post-processing (interpretation) of results.
- Improve control on out-of-vocabulary behavior.

A common requirement unifies these goals: Results should be independent of reference patterns or vocabulary [4]. This is the new requirement for word recognition which leads to the proposal put forward in this paper.

5. Definition of recognition scores

This section proposes a new way of reporting the results of word recognition that fulfills the requirement stated above. Two output values will be defined: one for 'quality', and one for 'similarity'. For each of these definitions, an objective function will be given. Later, in section 6, a performance metric for these result values will be defined that answers the question: How exactly do the values predict the correctness of results? This will enable comparison of dissimilar recognition engines. If different types of recognizers are to be combined, the type of results they produce has to be the same.

5.1. Two scores for recognition

All recognition engines are based on some sort of similarity metric, resulting in a list of result words matching the reference pattern to some degree and ordered by the similarity measure. Each word within the ranking is thus accompanied by a number specifying the quality of the match. But this match quality can be interpreted in two different ways. This can be explained by the characteristics of the tasks of writing and reading. See Fig. 1 for a visualization of the transformations taking place during this task. The two types of similarity can be interpreted as distortions arising when a writer's intention is transferred to paper (misspellings and individual writing style), but also vice versa (classification quality and interpretation by means of redundancy).

To illustrate the situation during recognition, let the best textual interpretation, or *nominal hypothesis*, be defined as the one virtual reference pattern which would create the best match within the recognition system. Then we can differentiate between similarity of nominal image input to nominal hypothesis, and similarity of nominal hypothesis to each reference pattern.

In any recognition system providing one single scalar value for valuation of results, this value has to combine the two mentioned similarity measures. However, the result of word recognition would be more useful if it consisted of two values: a similarity measure for each reference pattern, combined with a probability for the quality of recognition process itself:

- Similarity of match – corresponds to a 'dyslexic factor', because it can be interpreted as misspellings that have happened during the writing process and are corrected during recognition.
- Quality of recognition – corresponds to a 'scribble factor', because it can be interpreted as the conformity of writing style to styles known to the recognizer.

The interface is sketched in Fig. 2. In the next sections, we will define the two different result values.

5.2. Similarity measure

In traditional terms, the confidence value is an approximation to the probability of 'correctness' (probability that the image in fact has the meaning reported), where '1' denotes 'correct' and '0' denotes 'error'. Since correctness is not independent of the reference pattern, e.g. the size of the vocabulary, a function has to be approximated that interpolates the correctness function. A good candidate for this function is string similarity, with the same scale from '1', denoting 'exact match', to '0', 'completely wrong'. This similarity measure has its merits in different contexts:

- During training of the recognition engine, the adaptation to OOV words which should be rejected is eased by supplying the similarity.
- During recognition, reporting the similarity of reference patterns to recognition results circumvents the problem of premature decision between similar results.

At this point, any similarity function will do; as the simplest approach we choose the longest common subsequence (lcs) of two word-strings w_1, w_2, normalized by their length:

$$s_{\text{ref}} = d(w_{\text{nom}}, w_i).$$

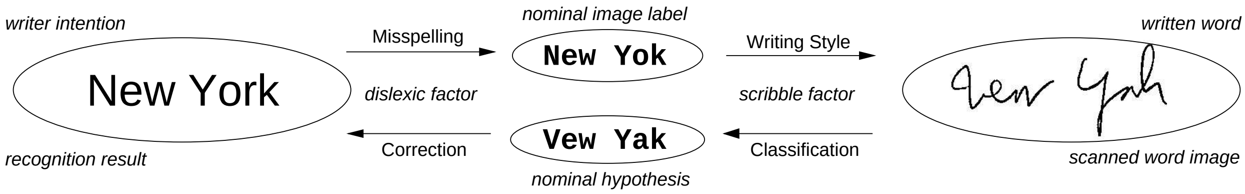

Figure 1. Two types of transformation during writing (top path) and recognition (bottom path)

For providing the similarity values, one needs to have a *nominal pattern*, w_{nom}, for comparison. It is extracted within the two different contexts mentioned above, training and recognition, situations which have to be distinguished:

- The *nominal image label* is determined during labeling of training data. It defines the objective function for similarity. Its specification is not as trivial as it may seem, because human readers always anticipate meaningful words and therefore always apply reference patterns implicitly. In case of easy-to-read patterns misspellings can be corrected, but in case of cursive script, assumptions about the context nearly always have to be made, thus obliterating the 'true' nominal pattern.
- The *nominal hypothesis* is determined during recognition, without access to the reference pattern. The recognition engine has to provide something like a best match at character level. This can be achieved directly in two-step processes (character recognition, then correlation), but is more difficult with integrated approaches like HMMs. The strength of these approaches lies just in their inherent error tolerance, which does not hypothesize a nominal result.

5.3. Quality measure

In the new interface, the readability of writing is quantified by a corresponding quality measure. Bear in mind that the quality of an image is a value clearly dependent on the recognition engine. For example, character based recognizers will see cursive script as having low quality, but special script recognizers will report high quality.

What would be a good objective function for the quality measure? A specification of quality during training could be done by calculating the string distance between nominal image label and nominal hypothesis.

$$q_{\text{ref}} = d(w_{\text{nom}}, w_{\text{hyp}})$$

At this point, circular reasoning appear to enter, as string distance has already be used for the definition of similarity. But in fact distances between different strings are being used as objective functions during training:

- For similarity estimation, the distances between nominal image label and reference patterns are used.
- For quality estimation, the distance between nominal hypothesis and nominal image label is used.

During recognition, no nominal image label is available, therefore no distances can be used for estimation.

5.4. Relationship to single confidence scores

The new measures which have to be determined by the recognition engine can be related to conventional word recognition. There, in terms of the new output values, the objective function for *similarity* to the nominal label is '1', for all other hypotheses is '0'; the objective function for *quality* of any recognizable image is '1', for unreadable images is '0'. Reject (OOV) training is performed when the objective function is '0'. A single traditional confidence score can be interpreted as the product of quality and similarity.

6. Performance metrics

The performance of word recognition is traditionally given with recognition and error rates, given in percentages. The proposed interface has consequences for this definition of recognition performance. One of the new requirements in section 4 is that result scores have to be independent of reference patterns. In some respect, this makes recognition 'easier':

- With traditional recognition engines, for each reference word, a hard yes-or-no decision with a confidence score has to be given in order to evaluate the engine.
- In similarity based recognition engines, for each reference word, only a soft similarity value – connected with a quality score – has to be given.

If no hard decisions have to be provided by the recognition system, neither recognition nor error rates can be reported and are not even defined, and therefore no direct correlations between error rate and recognition rate can be obtained. This makes a comparison to traditional systems difficult. But on the positive side, test-deck dependence is eased, and no special OOV tests have to be conducted.

There are two functions to evaluate: similarity and quality. As in traditional evaluations, we have a set of images

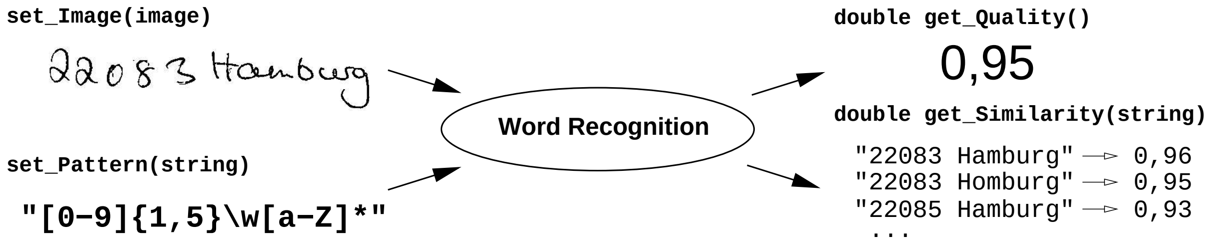

Figure 2. The proposed interface, providing two result values

containing data label which we compare against the values estimated by the recognition engine. Additionally, relevant reference patterns have to be provided, because *all* similarities have to be evaluated. This allows a test of reject (OOV) behavior in a natural way.

6.1. Performance based on correctness of similarity

The first evaluation examines the performance of the similarity measure. It is an evaluation of the power of the recognition engine itself. The difference to an evaluation of traditional systems is only the looser definition of 'correct'. We ignore the quality measure and evaluate only the performance of similarity results. The degree of *similarity correctness* C_s can be given as

$$C_s = 1 - (s_{\text{est}} - s_{\text{ref}})^2.$$

The negative term $(s_{\text{est}} - s_{\text{ref}})^2$ gives the deviation from the ideal correspondence between correct similarity s_{ref} (given by string similarity with the nominal image label) and the estimated value s_{est} (calculated by the recognition engine). This penalty can be interpreted the following way: In cases when reference similarity is clear, that means '0' or '1', the maximum penalty of '1' can be reached; in other cases, when a reference pattern has a 50 percent similarity, the maximum penalty is much lower. This behavior is useful, because it reflects the significance of errors. But it has to be kept in mind when interpreting the average error.

The performance P is given as an average over all patterns. It is the simplest measure, the mean square error:

$$P_{\text{engine}} = 1 - \frac{1}{N}\sum_{N}(s_{\text{est}} - s_{\text{ref}})^2.$$

To compare it with traditional systems, remember that hard recognition is *more* difficult when using this measure: It has to say 'no' to all images that are not exactly identical.

6.2. Predictability based on quality score

The performance evaluation of the quality measure corresponds to an evaluation of a confidence tagger. It answers the question, how well is the individual recognition performance, regarding the current recognition task, predicted by the quality value. The performance itself is given by the 'correctness of similarity' described in the section above.

In traditional systems, the confidence value has two, slightly differing tasks: First, it pretends to give the 'probability for true answers', forecasting the recognition quality. Secondly, it is meant to help improve the ratio of recognition to error rate. Different evaluation metrics are described in [7, 8]; these authors are interested mainly in the improvement of this ratio, examining the information gain by using additional features.

With the proposed interface, the second task of the confidence measure is fulfilled by the similarity value. Features used for estimating confidence are also used for similarity. Thus, when evaluating the quality score now, only the predictability is examined. Can the quality value be used to predict the likelihood that the estimated value of similarity is correct? As no probability for particular events like 'correct' can be given, the average deviation of estimated similarity from reference similarity will be predicted. This can be interpreted as probabilities for similarity ranges.

For mapping confidence scores to probabilities of correct answers, histogram approaches are often used [3]. For any confidence value, correct answers are counted. In our approach, the degree of similarity correctness C_s is mapped to the quality score q_{est}. For a qualitative analysis of the usefulness of the quality score, if a continuous slope in the histogram $C_s(q_{\text{est}})$ is obtained, this proves that confidence mapping, and thus quality prediction, is possible.

7. Calculation of recognition scores

Let us now consider how the two scores of a word recognizer, as defined above, can be estimated. Quality and similarity measure, illustrated as 'scribble' and 'dyslexic' factor, have to be extracted and separated from the various output values of the different recognition methods.

There are basically two possibilities to determine confidence values in traditional recognition systems: If the recognition engine produces a single score, an easy method for the estimation of the probability of correctness is confidence mapping. By this method, real probabilities can be assigned to recognition scores lying in an arbitrary range. But to achieve better results, often multiple features are used. A good introduction into the type of features used for estimating the confidence score is given in [6]: scores from first-best, second-best and n-best recognition are used, as well as those from garbage models etc. In this case, classifiers like neural nets are used to derive confidence values from feature sets.

But now calculation has to be independent of vocabulary [4]. Thus, second-best and n-best features cannot be used here. To compensate for these losses, additional features are needed. Available are the pattern itself and a recognition score, which is often complemented with additional information from the recognition process, like alignment and segmentation qualities, and recognition qualities of individual characters. A good addition would be a nominal hypothesis with all of the features mentioned above.

Again, two possibilities for the estimation of a quality value exist. Either heuristics are used to generate one single score from the given feature set, followed by a mapping to probabilities, or some sort of classifier combines the features. The first method will be used for quality estimation; features include a score for nominal recognition, detailed segmentation information and geometrical features, e.g. writing lines. Similarity is determined by the second method, a classifier which combines features like the score for the reference pattern and some sort of weighted edit distance (e.g. Levenshtein) to nominal recognition.

7.1. Character based word recognition

For hand block and machine print, word recognition engines most often are based on single-character recognition. They use scores of the individual characters and evaluate whole words e.g. by the Levenshtein distance. It's quite easy to implement the estimation for quality and similarity because, using the two-step approach, the respective features are already available:

- *Quality:* The average confidence c from nominal character classification can be used for quality estimation:

$$q_{\text{est}} = \text{avg}(c_i^{\text{nom}}).$$

- *Similarity:* The weighted edit sequence, with substitution weights corresponding to classifier results (distance to best result), and insertion and deletion costs k given by heuristics, are used to estimate similarity:

$$s_{\text{est}} = 1 - \frac{1}{n}\Big(\sum_{\text{sub}}(c_i^{\text{nom}} - c_i^{\text{ref}}) + \sum_{\text{ins}} k_{\text{ins}} + \sum_{\text{del}} k_{\text{del}}\Big).$$

7.2. Script word recognition, based on HMMs

For script word recognition the calculation of both values is slightly more difficult, because it is traditionally based on word lists and uses context knowledge extensively. Therefore, normally no nominal recognition is performed; instead, estimations for so called joker models are done. If the engine is based on HMMs, it provides a logprob p for any word hypothesis [5].

- *Quality* estimation can be done by joker model estimation. A mapping to the valid quality range can be performed: Other possibilities, like geometrical features, could also be used.

$$q_{\text{est}} = \text{map}(p^{\text{joker}}).$$

- *Similarity* can be estimated by the distance of logprobs between reference and nominal pattern, the latter given by the joker model. Again, it can be mapped to a valid range for similarities.

$$s_{\text{est}} = \text{map}(p^{\text{ref}} - p^{\text{joker}})$$

8. Summary and outlook

The traditional view of the output interface from word recognition has been described and its shortcoming have been analyzed. This has lead to the definition of a new interface for word recognition. Instead of decisions and confidences, word similarities are reported, and a quality measure has been assigned to the recognition process itself. This new view of word recognition raises the need for new performance metrics, which have also been proposed. Implementation for two different recognition engines have been outlined.

As next step, different recognition engines have to be evaluated, and their performance characteristics will have to be compared. Then, combination schemes for different recognizers must be developed to profit from the provided output.

References

[1] A. Brakensiek, J. Rottland, and G. Rigoll. Handwritten address recognition with open vocabulary using character n-grams. In *8th IWFHR*, pages 357–362, 2002.

[2] A. Brakensiek, J. Rottland, and G. Rigoll. Confidence measures for an address reading system. In *7th ICDAR*, pages 294–298, 2003.

[3] J. M. Gloger, A. Kaltenmeier, E. Mandler, and L. Andrews. Reject management in a handwriting recognition system. In *4th ICDAR*, pages 556–559, 1997.

[4] L. Jiang and X. Huang. Vocabulary-independent word confidence measure using subword features. In *5th ICSLP*, 1998.

[5] A. Kaltenmeier, T. Caesar, J. Gloger, and E. Mandler. Sophisticated topology of hidden Markov models for cursive script recognition. In *2nd ICDAR*, pages 139–142, 1993.

[6] J. F. Pitrelli and M. P. Perrone. Confidence modeling for verification post-processing for handwriting recognition. In *8th IWFHR*, pages 30–35, 2002.

[7] T. Schaaf and T. Kemp. Confidence measures for spontaneous speech recognition. In *ICASSP*, pages 875–878, 1997.

[8] M. Weintraub, F. Beaufays, Z. Rivlin, Y. Konig, and A. Stolcke. Neural-network based measures of confidence for word recognition. In *ICASSP*, pages 887–890, 1997.