**Marcin PŁONKOWSKI, Pavel URBANOVICH**

Katolicki Uniwersytet Lubelski Jana Pawła II, Lublin, Polska

# Tuning a CMU Sphinx-III Speech Recognition System for Polish Language

***Abstract.*** *In this paper, authors describe parameters which may be tuned to obtain the best performance and accuracy for a large vocabulary continuous speech recognition task. Behavior of certain parameters should be similar regardless of the language speech recognition. However, some parameters will have a different impact on the accuracy of the Polish speech recognition as compared to the English speech recognition.*

***Streszczenie.*** *W niniejszym artykule autorzy opisują paramenty, które mogą być dostosowywane, w celu uzyskania większej wydajności i dokładności w zadaniach rozpoznawania mowy ciągłej. Zachowania pewnych parametrów powinny być podobne bez względu na używany język. Jednakże niektóre parametry będą miały inny wpływ na dokładność rozpoznawania mowy polskiej w porównaniu do zadań rozpoznawania mowy angielskiej (**Strojenie systemu rozpoznawania mowy CMU Sphinx-III dla języka polskiego**).*

**Keywords:** speech recognition, CMU Sphinx, polish language.
**Słowa kluczowe:** rozpoznawanie mowy, CMU Sphinx, język polski.

doi:10.12915/pe.2014.04.42

## Introduction

CMU Sphinx-III is one of the most popular speech recognition systems [1]. It works very well in continuous speech recognition tasks with a lot of words, regardless of speaker. However, to achieve satisfactory results, system must be trained on the appropriate set of utterances with the reference transcription.

In addition to the training, important aspect is the appropriate tuning parameters of the decoder (speech recognition system), in this way to achieve pretty good results in a reasonable time. Sphinx system was frequently tested and analyzed in relation to the English language [2-4] and other languages [5, 6]. With regard to the Polish language publications can also be found using the system Sphinx [7]. However, there is lack of publications, that have made the analysis of the most important parameters responsible for the accuracy of speech recognition and performance time for the Polish language.

CMU Sphinx-III is a system that uses statistical methods. Namely, this system is based on a hidden Markov model (HMM). It is now the dominant solution for the most recently designed speech recognition systems. If we have a good learning set (of appropriate size and of appropriate quality) the system gives very good results (word error rate is approximately 15%).

To obtain very good results training set size should take into account the following recommendations:

- 1 hour of recording for command and control for single speaker,
- 5 hour of recordings of 200 speakers for command and control for many speakers,
- 10 hours of recordings for single speaker dictation,
- 50 hours of recordings of 200 speakers for many speakers dictation.

## Preparing data for CMU Sphinx

The authors undertook the task of preparing a set of utterances, train the system and tune the parameters in order to achieve the best results, both in terms of quality and efficiency of speech recognition for the Polish language.

As the training set were used read speech 4 speakers (2 women and 2 men). Utterances varied in length from about 1 second to about 1.5 minutes. In total there was 1460 utterances of the total length of 2 hours and 11 minutes. Dictionary has 5168 words.

Test set consists of a utterances 2 people (1 woman and 1 man). There were other persons and other utterances than those used in training set In total, test set contains 75 items with a total length of 8 minutes.

Each utterance was recorded with a sampling rate of 16000 Hz and mono channel with 16 bits per sample.

To prepare the system to work with the Polish language should make a few modifications. First, we decided to use the Unicode character encoding (which facilitates work with other languages than English). The second important element is properly defined set of phonemes. Polish language can distinguish about 40 phonemes. However, some phonemes are very rare. Therefore, the optimal set in this case is 35 phonemes found in our training set (a, ą, b, c, cz, ć, d, dz, dź, e, ę, f, g, h, i, j, k, l, ł, m, n, ń, o, p, r, s, sz, ś, t, u, w, y, z, ź, ż and the special phoneme SIL - indicating silence).

The third thing is well defined dictionary along with the record of every word in phonetic notation (with the use of specified above phonemes). For example, the word *chęci* in phonetic form looks like this - h e ń ć i.

As already mentioned our dictionary contains more than 5,000 words. Therefore, it was necessary to write a program to realize the transcription in an automatic way.

Fortunately the way of reading in the Polish language has quite clear rules. They are defined as follows. First we have defined exceptions. For example word *zamarzać* (eng. to freeze) in phonetic form looks as *z a m a r z a ć*, not *z a m a ż a ć*. Then we defined the rules of phonetic notation.

For example, the rules for the letter *d* may look like this:

1) AdA; A d; 2;
2) AdD; A d; 2;
3) AdB; A t; 2 .

This means that if the text is found in the following sequence (the first rule): a vowel, the letter *d* and a vowel, this should be replaced by a sequence of two letters (a vowel and the letter *d*). However, note that the third rule will change voiced phoneme d to voiceless phoneme t. This is so because after the phone *d* are voiceless consonant.

All existing phonemes are divided into the following sets:

- all letters - the set X = { "a", "ą", "b", "c", "ch", "cz", "ć", "d", "dź", "dż", "e", "ę", "f", "g", "h", "i", "j", "k", "l", "ł", "m", "n", "ń", "o", "ó", "p", "r", "s", "sz", "ś", "t", "u", "w", "y", "z", "ź", "ż" };
- all vowels - the set A = { "a", "ą", "e", "ę", "i", "o", "ó", "u", "y"};

- all consonants - the set Q = { "b", "c", "ch", "cz", "ć", "d", "dź", "dż", "f", "g", "h", "j", "k", "l", "ł", "m", "n", "ń", "p", "r", "s", "sz", "ś", "t", "w", "z", "ź", "ż" };
- fricatives - the set S = { "ch", "f", "s", "sz", "ś", "w", "z", "ż" };
- labial consonants - the set W = { "b", "m", "p"};
- palatal consonant- the set Ś = { "ci", "ć", "dzi", "dź", "ź" };
- alveolar consonants - the set Z = { "c", "cz", "d", "dz", "t"};
- velar consonants - the set T = { "g", "k", "u" };
- voiced consonants - the set = { "b", "d", "dz", "dź", "dż", "g", "j", "l", "ł", "m", "n", "r", "w", "z", "ź", "ż" };
- voiceless consonants - the set B = { "c", "ch", "cz", "ć", "f", "h", "k", "p", "s", "sz", "ś", "t" }.

Based on the rules and this sets, program has generated automatic transcription for each word defined in the dictionary. Then, for each recorded statement had to be prepared text in orthographic notation.

Every statement has been recorded in wave file format. Unfortunately, this form of presentation of sound recording does not work in speech recognition systems. Thus, the extraction procedure is required, to bring out the most desirable features.

The authors selected the most commonly used in these types of systems parameters - cepstral coefficients.

The procedure begins with the distribution of the speech signal into frames (with a length of 410 samples). Then, for each frame the following steps are performed:

1) Pre-Emphasis (boosting high frequencies),
2) Windowing (smoothes the edges of window),
3) FFT (Fast Fourier transform),
4) Mel-Filter Bank (warp frequencies from Hz to Mel frequency scale),
5) Log (logarithm),
6) $FFT^{-1}$ (Inverse Fourier Transform),
7) Deltas (first and second derivative).

Then we obtain 39 cepstral coefficients (for each frame). The prepared systems were trained Baum-Welch algorithm. It is the most popular training algorithm in systems based on hidden Markov model.

**Pruning parameters**

The whole process of speech recognition by decoder starts with acquisition of utterance. Then, the extraction process is performed of the most desirable features (from the point of view of speech recognition system). Decoder analyzes these features using acoustic model, language model and vocabulary. Block diagram is shown in fig.1.

In this article, we analyze the parameters of the model language. The other parameters remain unchanged and retain their default values.

Pruning parameters allow for the optimization of search algorithms by eliminating unlikely search paths. Of course, if parameters are highly pruned the gain in efficiency with compromising the recognition accuracy.

If, however, you set the parameters of the low degree of pruning it will increase the accuracy but at the cost of longer computation time. So the an important element is to find a compromise to pruning parameters, in order to maintain a high recognition accuracy, while a reasonable duration of the process.

Pruning parameters are: **-**beam, **-**pbeam i **-**wbeam [5]. Parameter **-**beam determines which HMMs remain active at any given point (frame) during recognition. (Based on the best state score within each HMM.) Parameter **-**pbeam determines which active HMM can transition to its successor in the lexical tree at any point. (Based on the exit state score of the source HMM.) Parameter **-**wbeam determines which words are recognized at any frame during decoding (based on the exit state scores of leaf HMMs in the lexical trees).

Tuning system begins with the selection of certain default parameters having to work properly in most situations. System Developers are proposing to set the initial values of parameters **-**beam and **-**pbeam on 1e-60 and -wbeam on 1e-30. Then increase or decrease the parameters in order to obtain the best results. Parameters -beam and -pbeam should be modified together. It was only after finding satisfactory results, modify the parameter -wbeam. Details of the algorithm can be found in [8].

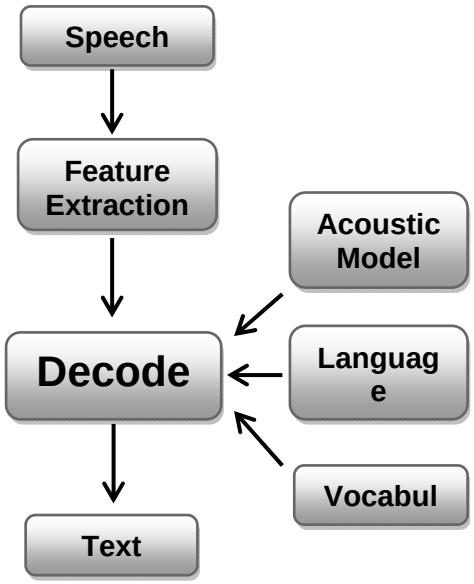


Fig.1. Block diagram of speech recognition system

The authors of this article decided to search all possible parameters combinations (with a certain interval) and with their participation evaluate the performance of the decoder. In addition to the accuracy of speech recognition was taken into account performance time. Parameters -beam and -pbeam take values from the set of {1e-80, 1e-70, 1e-60, 1e-50, 1e-40}. Whereas parameter -wbeam accepts values from the set of {1e-30, 1e-25, 1e-20, 1e-15, 1e-10, 1e-05}. Range of parameters has been selected by the preselection, thanks to which managed to identify the most promising areas. So, all options were 150.

For of each of them the decoder was running (on the test set described earlier). In order to evaluate performance time is worth complementing with information, that the tests were made on the quad-core processor (Intel Core2 Quad Q8300 @ 2.50GHz).

For evaluation were used two parameters: the recognition accuracy and the performance time.
We estimate the accuracy of using number of incorrectly recognized words WER (word error rate), which is defined as:

$$WER = \frac{S + I + D}{N}, \quad (1)$$

where: S is the number of substitutions, I is the number of insertions, D is the number of deletions, N is the number of words in the reference.

The word error rate (WER) is the most common way to evaluate speech recognizers. The word error rate is defined

as the sum of these errors divided by the number of reference words. It is worth noting that according to the formula (1) WER value may be greater than 100%.

When reporting the performance of a speech recognition system, sometimes word accuracy (WAcc) is used instead:

$$WAcc = 1 - WER. \quad (2)$$

Of course, our performance evaluation will be the better the lower the level of WER and the shorter will be the time of recognition. However, we decided to pay more attention to accuracy than the execution time, hence the quality of our performance evaluation results from the following formula:

$$Score = WER^2 \cdot Time \cdot 1000. \quad (3)$$

Constant 1000 is added only for easier presentation of scores in the table. Whereas time is presented in the format mi:ss. Because of the number of possible combinations in the table we present only a portion of the most significant results.

Table 1. Parameters evaluation: -beam, -pbeam, -wbeam

| -beam | -pbeam | -wbeam | WER | Time | Score |
|---|---|---|---|---|---|
| 1e-80 | 1e-80 | 1e-20 | 35.52% | 6:24 | 0.5609 |
| 1e-40 | 1e-40 | 1e-05 | 61.50% | 2:57 | 0.7748 |
| 1e-70 | 1e-50 | 1e-20 | 36.76% | 4:57 | 0.4644 |

The above table presents the three rows. The first has the lowest coefficient of WER in the entire test set. However, the performance time is approximately 6.5 minutes. The best time we can reach for the parameters presented in the second row. Operating time is less than 3 minutes. The most satisfactory solution, which has the lowest rate in the "Score", is located in the last row of the table. We see that the WER is still at the appropriate level (close to the best score), but definitely we were able to reduce the performance time by 1.5 minutes.

Below you will see three tables with charts that graphically illustrate the effect of different parameters (-wbeam, -pbeam, -wbeam) on the efficiency of speech recognition.

Table 2 shows the impact of parameter -beam on the efficiency of the speech recognition (at fixed values of the parameters: -wbeam and -pbeam).

Table 2. Evaluation parameter -beam

| -beam | -pbeam | -wbeam | WER | Time | Score |
|---|---|---|---|---|---|
| 1e-80 | 1e-50 | 1e-20 | 36.55% | 5:06 | 0.4731 |
| 1e-70 | 1e-50 | 1e-20 | 36.76% | 4:57 | 0.4644 |
| 1e-60 | 1e-50 | 1e-20 | 38.30% | 4:42 | 0.4787 |
| 1e-50 | 1e-50 | 1e-20 | 42.30% | 4:22 | 0.5426 |
| 1e-40 | 1e-50 | 1e-20 | 54.41% | 3:51 | 0.7916 |

The data in Table 2 are also presented in the graph (see Fig. 2).

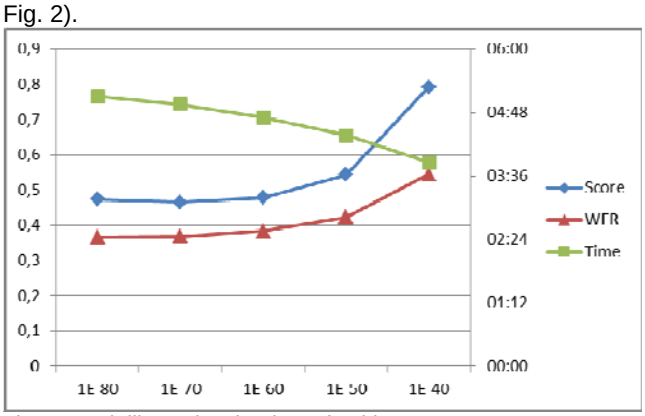


Fig.2. Graph illustrating the data of Table 2

Table 3 shows the impact of parameter -pbeam on the efficiency of the speech recognition (at fixed values of the parameters: -beam and -wbeam).

Table 3. Evaluation parameter -pbeam

| -beam | -pbeam | -wbeam | WER | Time | Score |
|---|---|---|---|---|---|
| 1e-70 | 1e-80 | 1e-20 | 36.04% | 6:06 | 0.5501 |
| 1e-70 | 1e-70 | 1e-20 | 36.04% | 5:46 | 0.5201 |
| 1e-70 | 1e-60 | 1e-20 | 36.65% | 5:20 | 0.4976 |
| 1e-70 | 1e-50 | 1e-20 | 36.76% | 4:57 | 0.4644 |
| 1e-70 | 1e-40 | 1e-20 | 41.58% | 4:32 | 0.5443 |

The data in Table 3 are also presented in the graph (see Fig.3)

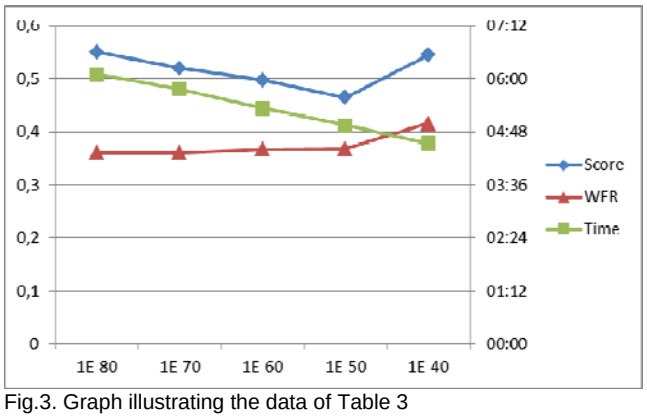


Fig.3. Graph illustrating the data of Table 3

Table 4 shows the impact of parameter -wbeam on the efficiency of the speech recognition (at fixed values of the parameters: -beam and -pbeam).

Table 4. Evaluation parameter -wbeam

| -beam | - pbeam | -wbeam | WER | Time | Score |
|---|---|---|---|---|---|
| 1e-70 | 1e-50 | 1e-30 | 37.06% | 5:10 | 0.4929 |
| 1e-70 | 1e-50 | 1e-25 | 36.96% | 5:11 | 0.4917 |
| 1e-70 | 1e-50 | 1e-20 | 36.76% | 4:57 | 0.4644 |
| 1e-70 | 1e-50 | 1e-15 | 37.99% | 4:56 | 0.4944 |
| 1e-70 | 1e-50 | 1e-10 | 39.22% | 4:34 | 0.4878 |
| 1e-70 | 1e-50 | 1e-05 | 47.13% | 4:03 | 0.6246 |

The data in Table 4 are also presented in the graph (see Fig. 4).

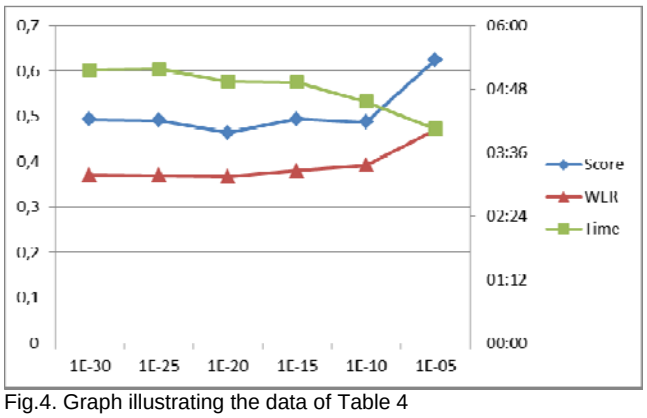


Fig.4. Graph illustrating the data of Table 4

Presented graphs show the effect of the parameters on the correctness of speech recognition and the duration of the algorithm. You have to be especially careful when choosing extreme values for each parameter, because it leads to a more rapid change in the evaluation of the model.

### Language Weight and Word Insertion Penalty

The next language model parameters are: -lw i -wip. Parameter -lw (language weight) can determine how language model affects the accuracy of speech recognition.

When evaluating hypothesis are taken into account the acoustic model and language model. However, this is not a simple product of these parameters. By -lw parameter we can determine the impact of the language model to evaluate the hypothesis. Parameter -wip (word insertion penalty) is a parameter specifying how great will be the "penalty" for the new word in a given hypothesis. Also, the choice of these parameters was analyzed. The values for the parameter -lw were selected in the range from 0.5 to 9.0 with step of 0.5. Whereas parameter -wip accepts values from the set of {0.01, 0.05, 0.1, 0.15, 0.2, 0.25, 0.30, 0.35, 0.40, 0.45, 0.5}. Evaluation of the quality was performed in accordance with the formula (2). The following table presents the main results.

Table 5. Parameters evaluation: **-**lw, **-**wip

| **-**lw | **-**wip | WER | Czas | Ocena |
|---|---|---|---|---|
| 9 | 0.05 | 36.24% | 4:59 | 0.4546 |
| 7 | 0.05 | 33.47% | 5:14 | 0.4071 |
| 5 | 0.05 | 33,98% | 5:31 | 0.4423 |
| 3 | 0.05 | 34,39% | 5:51 | 0.4805 |
| 1 | 0.05 | 35,83% | 6:08 | 0.5468 |
| 0.5 | 0.05 | 42.30% | 6:09 | 0.7642 |

The data in Table 5 are also presented in the graph (see Fig.5).

The above table shows only a few most important information. Namely, the first row shows the system parameters which help to achieve the best execution time. The second row shows the parameters for who achieved the most favorable result. The last row shows the results for the worst set of parameters. Thus it is clear that by inappropriate selection of parameters can significantly degrade the recognition result as well as its duration.

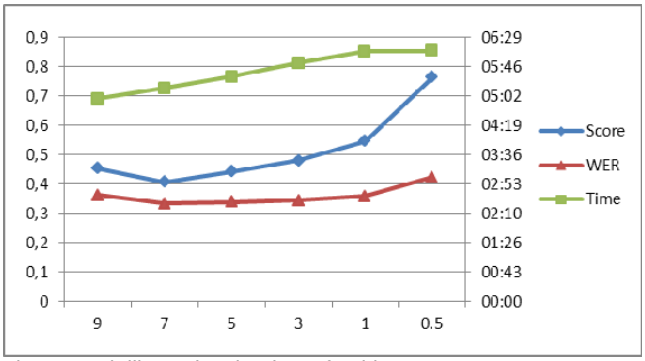


Fig.5. Graph illustrating the data of Table 5

### Conclusions

The purpose of this article was to examine what influence on the quality of recognition has selection of the most important parameters of the system. Based on the data presented, it is clear that in addition to the correct train the model, it is necessary to tune the decoder. Comparing randomly selected parameters for the most efficient, it turns out that the error rate can be significantly reduced.

Additionally was analyzed execution time, which is also a very important parameter. It is therefore important to find the right balance between recognition accuracy and execution time. That's all we can achieve by proper selection of the parameters that must be found experimentally. This is connected with the need to adjust system parameters for speech recognition and linguistic characteristics of a given set are working with. Therefore, it turns out that for other languages and even for other training samples, the corresponding parameters may vary.

### REFERENCES


[1] CMU Sphinx: The Carnegie Mellon Sphinx Project. http://cmusphinx.sourceforge.net/, Apr 2013

[2] Vertanen K., Baseline WSJ Acoustic Models for HTK and Sphinx: Training Recipes and Recognition Experiments, Cavendish Laboratory, University of Cambridge, 2006

[3] Novak J., Dixon P., Furui S., An Empirical Comparison of Sphinx and HTK models for Speech Recognition, in Proc. ASJ 2010, pp. 73-74, Mar. 2010

[4] Vertanen K., CMU Sphinx Wall Street Journal Training Recipe, http://www.inference.phy.cam.ac.uk/kv227/sphinx/, 2006

[5] Varela A., Cuayuhuitl H., Nolazco-Flores J.A., Creating a Mexican Spanish Version of the CMU Sphinx-III Speech Recognition System, CIARP 2003, LNCS 2905, 251-258, (2003)

[6] Satori H., Harti M., Chenfour N., Arabic Speech Recognition System Based on CMUSphinx, IEEE Proceedings of ISCIII'07, Morocco, 31-35, (2007)

[7] Janicki A., Wawer D., Automatic Speech Recognition for Polish in a Computer Game Interface, In proceeding of: Federated Conference on Computer Science and Information Systems - FedCSIS 2011, Szczecin, Poland, 18-21 September (2011), 711-716

[8] Sphinx-3 s3.X Decoder, http://www.cs.cmu.edu/~archan/s_info/Sphinx3/doc/s3_description.html, Apr 2013.



***Authors:***
*dr Marcin Płonkowski; prof. dr hab. Pavel Urbanovich, Katolicki Uniwersytet Lubelski Jana Pawła II, Instytut Matematyki, Katedra Systemów Operacyjnych i Sieciowych, ul. Konstantynów 1H, 20-708 Lublin, E-mail: marcin.plonkowski@kul.lublin.pl,*