Language modeling using SOM network

Leszek Gajecki[1], **Ryszard Tadeusiewicz**[2]

[1]University of Information Technology and Management, ul. Sucharskiego 2, Rzeszów, Poland
[2]AGH University of Science and Technology, ul. Mickiewicza 30, Kraków, Poland
[1]lgajecki@wsiz.rzeszow.pl; [2]rtad@agh.edu.pl

Abstract

Trigram language model and its derivative are commonly used for English Language for Large Vocabulary Continuous Speech Recognition systems. However there is need of another model for languages with less strict word order (Slavonic languages). Authors work on language model, which could be good suited for Polish language. The idea is to use of neural network- Kohonen's Self Organizing Memory (SOM).

Keywords: Vocabulary Continuous Speech Recognition, Neural networks, Language Modeling

1. Introduction

The general aim of this paper is presentation of new approach to language modeling for Large Vocabulary Continuous Speech Recognition systems (LVCSR). We describe models based on Neural Networks – Self Organized Maps (SOM). This form of models could be more suitable for Slavonic languages with less strict words order than in English. We take into account Polish language, but some results can be generalized also for other Slavonic languages.

We perform recognition task using proposed model on speech recognition system, to compare it with widely used trigram models.

2. Related work

Language model module (LM) plays important role in LVCSR systems based on Hidden Markov Models framework. It gives score to considered words hypothesis, evaluating them according to language rules. These hypotheses are built by search engine to match spoken utterances. Without use of language model we get searching tree with large branching factor, but LM let for cutting down many of such branches, limiting search space. Examples of LVCSR systems could be HTK (Young et al., 2006), ESAT Speech Recognition System (Duchateau 1998; ESAT-PSI, 2006), or LVCSR systems for Polish Language (Szymański et al., 2008; Hnatkowska and Sas, 2008). All above systems use trigram language models, about which we will speak later.

Also Finite State Grammar (FSG) gives good enough results (Brocki and Korzinek, 2008). The drawback is that such grammar need to be constructed manually, which gives ability to use it for dialog systems where grammar is limited. In systems where we expect free form of speech limited only in vocabulary and natural language rules this approach require huge amount of work to construct FSG.

Very often used Trigram model, (Jelinek, 1997) is simple idea and very fast to evaluate word sequence, which is needed due to big amount of hypothesis obtained by search engine. Secondly it lets for partial generation of hypothesis and prune low scored one enough early, without search tree full expansion for that one. This is because of work on 3 last words. However the data sparseness is problem which is partially solved by its derivative models like class-based trigram model (Brown et al., 1992), Knesser - Ney class-based model (Knesser, Ney 1995), back off trigram model, count – based interpolated LM with Jelinek-Mercer or Chen and Goodman's smoothing (Chen, Goodman 1998).

Another property of such models is that they assume strict word order, which suits them for English, but not for Slavonic Languages, where such order in short word span often is freer. Allowed move of words without changes in their meaning produce sentences which mostly could differ only in style of speech and maybe putting stronger accent on some parts of the sentence. For example sentence "*Ona widziała jego*" (In Polish: She saw him), we can express on following five ways (while its English versions not always are correct).

Ona jego widziała. (She him saw).
Jego ona widziała (Him she saw).
Jego widziała ona.(Him saw she).
Widziała jego ona. (Saw him she).
Widziała ona jego. (Saw she him).

This short example show that learning set should include rewritten sentences in all allowed versions, what is mostly not possible.

Weighted Finite State Transducer considered in Speech Recognition (Benesty et al., 2008), or connectionist Neural Network as technique for n-gram weight smoothing (Bengio 2006; Xu and Rudnicky 2006) base also on words order. We need to mention that last idea was also applied to Polish Language (Brocki, 2010).

In previous works (Gajecki and Tadeusiewicz 2008) we considered use of simple Head-Driven Phrase Structure Grammar (HPSG) described as constraint- based formalism. It consists of small number of general rules, and lexical entries, which describe words-specific dependencies. Because it is very strict formalism we need enough complex grammar, which require large amount of work to prepare lexicon. Therefore we search for other models too.

The discussion about some language models we presented in work (Gajecki and Tadeusiewicz 2009b). General information about Speech Recognition are described in (Benesty et al., 2008; Markovitz, 1996;

Duchateau, 1998). Early publication about Polish speech signal is (Tadeusiewicz, 1988).

3. Language model with Neural Networks application.

3.1. Introduction.

Artificial Neural Networks was used for speech recognition for acoustic modeling (Tadeusiewicz 1994; Robinson 1994). Their application in language modeling is not often, but there exists mentioned above connectionist Neural Network model (which use one or more layer of feed forward network). Another approach is to use recurrent network – Long-Short Term Memory (Brocki, 2010).

We present application of Kohonen's Self Organizing Maps for general grammar rules discovery. This work is continuation of our previous works (Gajecki, Tadeusiewicz 2010; Gajecki, Tadeusiewicz 2009a). Here we present results obtained on LVCSR system, but in that works after construction LM we preformed simplified recognition task. We developed also neural network model.

3.2. Application of Neural Network

Our aim is to obtain language rules automatically – association between classes of words, so we need network with unsupervised learning. We choose Kohonen's Self Organizing Maps (Kohonen 2001). Such network consist one layer of neurons. The inputs have to be normalized:

$$\|x\| = 1 \quad \text{with norm:} \quad \|x\| = \sqrt{\sum_{i=1}^{M} x_i^2} \qquad (3.1)$$

The outputs of neurons are computed:

$$y_j = \sum_{i=1}^{M} x_i w_{ij} \qquad (3.2)$$

Learning rule (Oja rule) is given by the equation:

$$\Delta w_{ij} = y_j (x_i - w_{ij}) \eta \alpha(t) h(j, win) \qquad (3.3)$$

where: w- weights, x- input, y- output, η - learning rate

$\alpha(t) = \frac{\alpha_0 t}{C+t}$ - diminishing function depended on time of learning (t-number of learning step), which is necessary to make learning stable

$h(j, win) = e^{-|j-win|}$ - neighborhood function (win is number of winner neuron, j – number of neuron, linear structure)

Our network will learn relationship between each two neighbor words in sentence. Input vector consist on sequence of 0/1 responsible for each property coding. Respectively – for first word in relation and second word. Each one is coded by code 1 from n, with additional position 0/1 indicating if given property is not present (value null). For example noun in Polish language has flexem, number, gender, case, but no person or degree. (In other languages can be different, like Turkish languages where nouns have personal and possessive affixes). In this way of coding we can notice, that input vector has always the same length (=56), because in each property we have one 1, the other values are only 0. In this way we don't need normalization.

Fleksem	Number	Gender	Case	Person	Degree
0 1 0... 0	0 1 0	0 1 0	0 1 0 0 ...	0 0 0 1	0 0 0 1

Fig. 1. Part of Speech (POS) coding

In our previous work we give sequence of zeros for that property, which not exist for given word. In case of learning we don't compute the difference for that property. It wasn't enough good solution, because we make influence to learning value 0 for such coordinates. Better solution is giving new element of vector for indicating that such property doesn't exist.

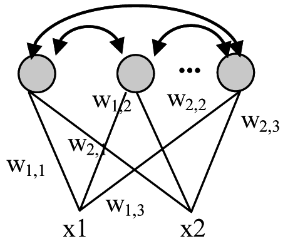


Fig. 2. Kohonen's Self Organizing Map

Network is trained using competitive learning – Winner Takes Most. We chose this method because of its better stability than Winner Takes All. During learning after computation of *y* the competition between neurons is performed. Neuron, which y value is highest means that the vector of this neuron's weights is closest to input vector. Such neuron is winner and its weights obtain the highest change their values, impact on neighbors is smaller (we have linear neighborhood structure).

Further information about neural network is described in (Wu Chou, Biing Hwang Juang 2003, Kohonen 2001, Duch et al., 2006). Book (Tadeusiewicz et al., 2007) can give introductory point of view for this field.

3.3. Language Rules

SOM network as clustering technique can group relations between words in clusters (fig. 3a). However we need to separate relations existing in language from those which doesn't exist. At the end of learning we will examine all training data again – how much data belongs to clusters represented by each neuron (coverage). These neurons which are associated with small number of data (below given threshold) – will be treated as negative rule. The rest neurons will be positive rules. Given input vector (properties of two words) such network will examine if exist relation between such two words (fig. 3b). Such threshold we call as *minimal coverage threshold*.

3.3.1. Simple SOM

Next subject is question which kind of relation we can discovery. First idea is simple– we put into input of network properties two neighbor words (*Simple SOM*), as shown on fig. 4a. The example of sentences in which such network can find relations is presented on fig. 5a. In assumption of enough number of similar relations in training set network can find relation between (noun, nominative, fem.) and (verb, 3rd person, singular). Depends on other sentences in training set and position of clusters such relation can be a bit weak like relation between (noun, nominative) and (verb, 3rd person).

This solution is kind of word-class bigram and in the same way as bigram or trigram model is sensitive to word's order, which behavior we want to avoid.

3.3.2. Binary relation SOM

Second solution is to give at the input of network properties of present word and m-th word before. Next we take such words in reverse order (*Binary Relation SOM*), architecture shown on fig. 4b. In such way we include all binary associations in given sentence. Learning them in reverse order can create rules which don't take into account if words appear in reverse order. Example of learned rules is presented on fig. 5b.

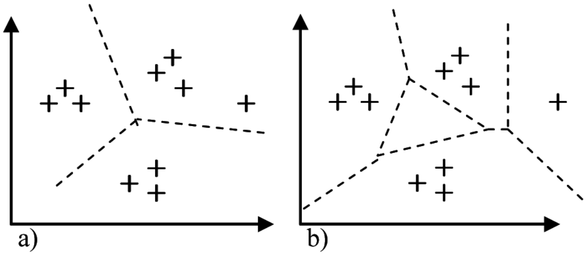
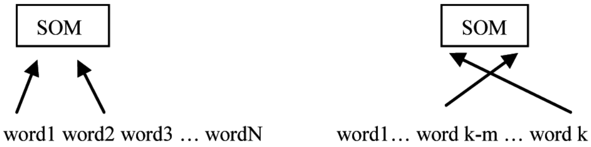
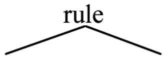


Fig. 3. Use of clusters versus use of positive and negative rules. Space divided into:
a) clusters each represent some points of data
b) clusters of positive rules (language rules) and negative clusters (no language rules)

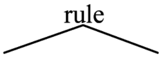


a)SimpleSOM network b)Binary Relation SOM
Fig. 4.Neural networks



Ona	widziała	jego
noun,nominativ, fem.,sing.	verb,3rd person,sing.	preposition,dativ

a) In Polish: She saw him



Babcia	ma	kota
noun,nominativ, fem.,sing.	verb,3rd person,sing.	noun,dativ

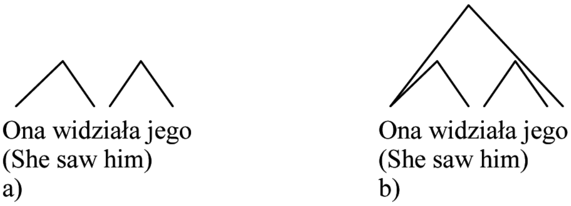
b) In Polish: Grandmother has a cat

Fig. 5. Example of rule that can be learned by SOM network.

3.3.3. Pair SOM

We extend our model to all possible word pairs from whole sentence. Such model will cover each relation between any two words (also in reverse order too). It let for all possible word order as we describe in chapter 1. Fig. 6 compares places all relation founded in sentence by this network and Binary Relation SOM, with examples of sentences accepted by such networks.

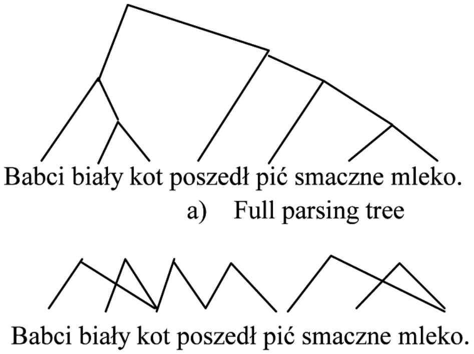
Described solution will parse one level of relations and not require that the whole sentence has to be parsed (in similar way to shallow parsers). Because most relationship in sentence appears between words which are not distant we limit maximal analyzed word span to M words before given word. This solution can also save searching tree expansion which is needed for whole sentence evaluation. Parsing tree of example sentence (fig. 7a), shows the distance in which works relationships in sentence. Fig. 7b shows one-level relations, which should be parsed by our model. When we choose word span M=2 this model became Binary Relation SOM described before.



c) *Ona widziała jego (She saw him)*
Widziała ona jego (Saw she him)

d) *Ona widziała jego (She saw him)*
Ona jego widziała. (She him saw).
Jego ona widziała (Him she saw).
Jego widziała ona.(Him Saw she).
Widziała jego ona. (Saw him she).
Widziała ona jego. (Saw she him).

Fig. 6. Comparison of networks, rule coming from fig. 5a. All relation applied by a) Binary Relation SOM and b)Pair SOM, both in forward and reverse order. Examples of another words order for sentence *Ona widziała jego (She saw him)* accepted by c) Binary relation SOM and d) Pair SOM,


Babci biały kot poszedł pić smaczne mleko.
Babci biały kot poszedł pić smaczne mleko.

a) Full parsing tree

b) One-level parsing tree

Fig. 7. Parsing trees for sentence (in Polish) Grandmother's white cat went to drink tasty milk

4. Experimental system

Our language models works on word lattices obtained by Speechlabs ASR (Szymański et al.; 2008, Demenko *et al* 2008). For lattice generation as partial results of speech

recognition there were used recordings of court decisions read or dictated by judges – about 4400 words, 388 sentences. The vocabulary has about 60000 words. Lattices were generated respectively using 4 tokens with unigram probabilities of these words.
For lattice decoding we use SRILM (Stolcke 2002) with necessary modifications. For our previous works we wrote our language model software in Java, so modified tool *lattice-tool* loads Java classes. For each word and its context evaluation it calls respective Java methods. Additionally we use also bash and gawk scripts. To accelerate our computation we plan to use CUDA in future.

4.1. Language models training.

For training of language model we use IPI PAN Corpus of Polish Language (Przepiórkowski 2004). Because of long amount of time required for neural network learning we process small part of this corpus – 600 000 words, 85 000 words lexicon. Computations were performed on PC with Pentium Dual Core processor 2,2 GHz, 3GB RAM.
We compare following models – respectively: bigram and trigram model Chen and Goodman's modified Knesser-Ney discounting Further we call them briefly as *2-gram, 3-gram Knesser-Ney*. Knesser-Ney model was chosen to comparison in works (Bengio 2006;Xu, Rudnicky 2006), because it gives very good results.
We perform experiment using bigger model to comparison: 3-gram model with modified Knesser-Ney discount, trained on 16 million words, gives WER=58,1%, computation time about 70min.
Our neural network has 100 neurons, we perform single iteration on mentioned text corpus (600 000 words, 85 000 different words,4 900 000 word-pairs by maximal word span 4 words. Coverage was computed after learning, for whole training set once again. Below we compare *Pair SOM* models and respectively word span: M=2,3,4 (*NN, max_wordspan =2,3,4*). To remind: word span M=2 means that model is *Binary relation SOM*

Model	WER	time
3-gram Kneser-Ney	62,4	11m37
2-gram Kneser-Ney	62,6	10m31
NN, max_wordspan =2	67,8	77m34
NN, max_wordspan =3	69,6	135m52
NN, max_wordspan =4	70,3	600m

Table 1. Word error rate (WER) for considered models.

MinCov	% of neurons accepted
1	78
770	50
16000	25
100000	10

Table 2. Levels of thresholds.

Next we made investigation about threshold of decision which neuron will be positive rule – minimal coverage threshold. Fig. 8 shows sorted coverage of SOM neurons, and level of considered threshold. According of this statistic we can check different levels of threshold. Their respective number of neurons being accepted as positive rules we put on Table 2. The results of above threshold applications we present in table 3.

Model NN	WER			
max_wordspan	minCov=1	770	16000	100000
2	68,0	69,5	**66**	68
3	72,1	71,2	71,2	**67,4**
4	73,8	71,5	70,2	**67,1**

Table 3. The influence of minimal coverage, small test set (280 words)

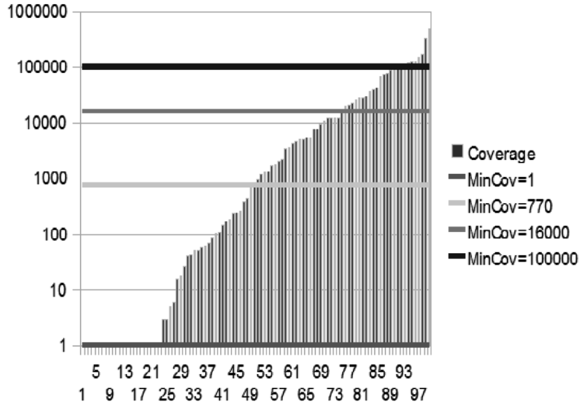


Fig. 8. Coverage of SOM neurons and levels of threshold.

4.2. Discussion.

We can notice that presented model is still not as good as bigram or trigram Knesser-Ney models. We didn't investigate into sub-optimality of learning parameters and number of neurons, so it can be the way of improvement. We expected that relation in longer word span could give better WER. Results of recognition of bigger set of sentences (tab. 1) don't prove that, however results of recognition small set (tab. 3) could support this idea. We see also, that such high coverage threshold, which cause acceptation small percent of neurons (here 10%), gives better results.
Computation time is much longer; however rewriting language model in C++, instead of calling Java from C++ (for each hypothesis) can accelerate computation. It is also interesting, that compared language Knesser-Ney 3-gram from bigger word corpus is computed longer too, while neural network gives response in the same time – dependent on network structure, not learning set.

5. Conclusions and future work

We presented new idea of language modeling, which go behind n-gram modeling. Results of experiments still didn't present that our model has better performance than chosen bigram and trigram model. However they show the need of further research on such model, which we plan to perform on neural network size and learning. We also plan to rewrite time-critic parts of software to C++ to reduce overhead for calling Java from C++, and also we want to rewrite some of them to CUDA or OpenCL.

Acknowledgements

Presented here research was supported by Polish Ministry of Science and Higher Education – PhD grant number N N516 513439. Computer used for this work was funded by Marshal Office of Małopolska voivodship. We would thank also to Laboratory of Speech Technology and Language in Poznań for computation of word lattices.

References

Benesty, J. Sondhi, M. M., Huang, Y. (2008), *Springer Handbook of Speech Processing,* Springer-Verlag, Berlin, Heidelberg.

Bengio,Y., Schwenk, H.,Senécal1,J.,S., Morin, F., and Gauvain, J.,L.,(2006) *Neural Probabilistic Language Models* in Holmes,D.,E., Lakhmi,C.,J., Innovations in Machine Learning - Theory and Applications, Book Series: Studies in Fuzziness and Soft Computing, Springer-Verlag, Berlin, Heidelberg.

Brocki, Ł,Korzinek, D. (2008), Marasek, K.,Telephony Based Voice Portal for a University, *Speech and Language Technology,* Vol. 11,pp 55-58.

Brocki, Ł. (2010), *„Koneksjonistyczny Model Języka w Systemach Rozpozanwania Mowy"* (In Polish *Connectionist Language Model in Speech Recognition Systems),* PhD thesis, Warszawa

Brown, Peter F., Pietra, Stephen A. Della, Pietra, Vincent J. Della, Lai, J. C., Mercer, Robert L., Class-based n-gram models of natural language, *Computational Linguistics* 18 (4), 1992.

Chen, S., F., Goodman, J., (1998) *An Empirical Study of Smoothing Techniques for Language Modelling,* TR-10-98, Computer Science Group, Harvard Univ.,

Demenko, G., Grocholewski, S., Klessa, K., Wagner, A., Ogórkiewicz, J., Lange, M., Śledziński, D., Cylwik, N.,(2008) *Jurisdic -- Polish Speech Database for taking dictation of legal texts.* Proc. LREC Conference, Marrakech, Morocco

Duch, W., Korbicz, J.,Rutkowski, L., Tadeusiewicz, R. (2006), (eds.) Vol. 6: Sieci Neuronowe (In Polish; *Neural Networks*) , in: Nałęcz, M. (eds.) *Biocybernetyka i inżynieria biomedyczna 2000*;(in Polish *Biocybernetics and biomedics engineery*) Exit,PTSN

Duchateau, J., (1998) *HMM based acoustic modeling in large vocabulary speech recognition*, PhD thesis, Katholieke Universiteit Leuven, Belgium http://www.esat.kuleuven.be/psi/spraak/.

ESAT-PSI (2006) *Description of the ESAT speech recognition system January 2006*, PSI - Speech Group, Katholieke Universiteit Leuven, Belgium. http://www.esat.kuleuven.be/psi/spraak/

Gajecki, L., Tadeusiewicz, R. (2008), *Modeling of Polish language for Large Vocabulary Continuous Speech Recognition* in Speech and Language Technology. Volume 11. Ed. Demenko, G., Jassem, K., Polish Phonetic Association, Poznań.

Gajecki, L., Tadeusiewicz, R., (2009a) Complex SOM network for Language Modelling in LVCSR - *Proceedings of 4th Language and Technology Conference* -Poznań 2009

Gajecki, L., Tadeusiewicz, R., (2009b) Language modeling and Large Vocabulary Continuous Speech Recognition, *Journal of Applied Computer Science*, vol.2 /2009, Łódź

Gajecki, L., Tadeusiewicz, R., (2010) *Modelowanie języka polskiego z wykorzystaniem gramatyki struktur frazowych.* (In Polish: *Modeling of Polish language using Head-Driven Phrased Structure Grammar*), in: Pohl, A., Goc, M., Konik, T., Siedlecka, M., (eds.), Rocznik Kognitywistyczny – Vol. III/2009, Wydawnictwo Uniwersytetu Jagielońskiego, Kraków 2010, pp. 59 – 68

Hnatkowska, B., Sas, J.,(2008) A*pplication of Automatic Speech Recognition to medical reports.,* Journal of Medical Informatics and Technologies, Vol. 12.

Jelinek, F., 1997. *Statistical Methods for Speech Recognition.* MIT Press, Cambridge, MA.

Kohonen, T. (2001). *Self-Organizing Maps.* Third, extended edition. Springer.

Kneser, R., Ney, H., (1995), *Improved backing-off for M-gram language modeling*' Proc. ICASSP, 181-184, 1995.

Markowitz, J. A. (1996), *Using speech recognition*, Prentice Hall PTR.

Przepiórkowski A. (2004) Korpus IPI PAN. *Wersja wstępna.* Instytut Podstaw Informatyki PAN, Warszawa, http://korpus.pl.

Robinson, A., J. (1994), *An Application of Recurrent Nets to Phone Probability Estimation,* IEE Transaction on Neural Networks, Vol. 5, No. 2,March, pp. 298-304

Szymański , M. ,Ogórkiewicz, J., Lange, M., Klessa, K., Grocholewski, S., Demenko, G. (2008), *First evaluation of Polish LVCSR acoustic models obtained fom the JURISDIC database*, Speech and Language Technology, Vol. 11,

Stolcke, A., (2002) *SRILM - An Extensible Language Modeling Toolkit*, in Proc. Intl. Conf. Spoken Language Processing, Denver, Colorado

Tadeusiewicz, R., (1988) *Sygnał mowy,*(in Polish *Speech signal*) Warszawa, Wydawnictwa Komunikacji i Łączności.

Tadeusiewicz R. (1994) *Zastosowanie sieci neuronowych do rozpoznawania mowy,* (in Polish *Neural Network Applications in Speech Recognition*) in Richter L. (eds.): Analiza, Synteza i Rozpoznawanie Sygnału Mowy dla Celów Automatyki, Informatyki, Lingwistyki i Medycyny, Polska Akademia Nauk, IPPT, Warszawa 1994, pp. 137-151

Tadeusiewicz, R., Gąciarz, T., Borowik, B., Leper, B. (2007) *Odkrywanie właściwości sieci neuronowych przy użyciu programów w języku C#* (in Polish *Neural networks properties discovery using C# programs*), Wydawnictwo Polskiej Akademii Umiejętności, Kraków

Wu Chou, Biing Hwang Juang(eds.) (2003), *Pattern Recognition in speech and language processing,* Boca Raton : CRC Press, 2003.

Xu,W., Rudnicky, A., (2006) *Can Artificial Neural Networks Learn Language Models?*

Young, S., Evermann, G., Kershaw, D., Moore, G., Odell, J., Ollason, D., Povey, D., Valtchev, V., Woodland, P., (2006) *HTK Book,* Cambridge University Engineering Department.