

# Text Conditioning and Statistical Language Modeling for Romanian Language

DOMOKOS JÓZSEF<sup>1,2</sup>  
TODERAN GAVRIL<sup>2</sup>  
BUZA Ovidiu<sup>2</sup>



SAPIENTIA  
University

**SpeD 2009**

Constanța, Romania  
June, 18-21, 2009



**TECHNICAL  
UNIVERSITY  
OF CLUJ-NAPOCA**

<sup>1</sup>SAPIENTIA UNIVERSITY, TG-MURES  
<sup>2</sup>TECHNICAL UNIVERSITY OF  
CLUJ-NAPOCA

*SpeD2009 – June, 18-21, 2009*

# CONTENTS

- Introduction
- N-gram language models
- Unigram, bigram, trigram language models
- Probability estimation and smoothing
- Good – Turing estimator
- Back – off smoothing
- Language model evaluation (Perplexity)
- Text conditionings
- Romanian Constitution corpus
- Experimental results
- Conclusions
- References

# INTRODUCTION

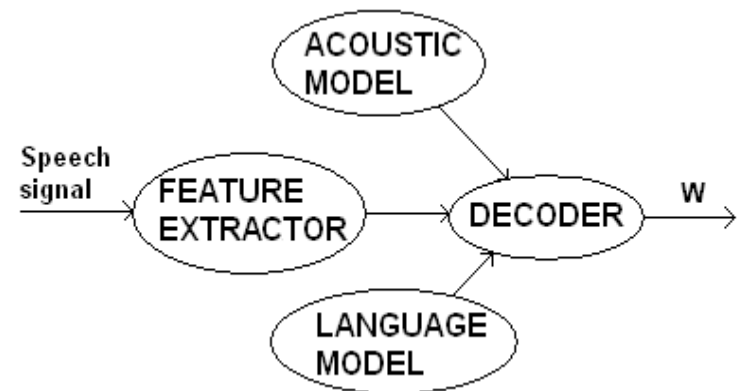
- A state of the art statistical speech recognition system is based on Hidden Markov Models (HMM's)
- Mathematically such a system can be described as follows: given a set of acoustic vectors (A), after the feature extraction stage, we are searching for the most probable word sequence (W).

$$W^* = \underset{w}{\operatorname{argmax}}\{P(W | A)\}$$

- The above equation can be transformed using Bayes rule as follows:

$$W^* = \underset{w}{\operatorname{argmax}}\{P(A | W) \cdot P(W)\}$$

- In this equation the probability  $P(A|W)$  represents the acoustic model and  $P(W)$  represents the language model part of the system



# INTRODUCTION

- We have developed so far the acoustic modeling part of the ASR using artificial neural networks [4]
- The scope of this paper is to present the language modeling part of the ASR and the text conditioning tools we have developed

# N-GRAM LANGUAGE MODELS

- The language model must estimate the probability of a word sequence  $w_1^n = (w_1, w_2, \dots, w_n)$  :

$$p(w_1^n) = p(w_1) \cdot \prod_{k=2}^n p(w_k | w_1^{k-1}) \quad \text{Eq. 1}$$

- The n-gram language models try to estimate the probability of the next word based on the *history* (the last  $n-1$  preceding words) [5] [7] [8] [9].

$$p(w_k | w_1^{k-1}) \approx p(w_k | w_{k-n+1}^{k-1}) \quad \text{Eq. 2}$$

- For  $n = 1 \dots 3$  we have:
  - Unigram language model ( $n=1$ )
  - Bigram language model ( $n=2$ )
  - Trigram language model ( $n=3$ )

# UNIGRAM, BIGRAM, TRIGRAM LM

- Unigram

- no history information is involved

$$p(w_k | w_1^{k-1}) \approx p(w_k)$$

$$p(w_1^n) = \prod_{k=1}^n p(w_k)$$

- Bigram

- takes in consideration one word for history

$$p(w_k | w_1^{k-1}) \approx p(w_k | w_{k-1})$$

$$p(w_1^n) = p(w_1) \cdot \prod_{k=2}^n p(w_k | w_{k-1})$$

- Trigram

- uses a two word history

$$p(w_k | w_1^{k-1}) \approx p(w_k | w_{k-1}, w_{k-2}) = p(w_k | w_{k-2})$$

$$p(w_1^n) = p(w_1) \cdot p(w_2 | w_1) \cdot \prod_{k=3}^n p(w_k | w_{k-1}, w_{k-2})$$

# PROBABILITY ESTIMATION AND SMOOTHING

- The probabilities for LM can be simply calculated using MLE (Maximum Likelihood Expectation) algorithm, but this method does not provide useful results until they are not smoothed
- Smoothing means that a probability mass is retained from high probabilities to be reallocated to zero or small probability values
- We have implemented add-one smoothing, Witten-Bell discounting and back-off smoothing
- For the experiments we used unigram, bigram and trigram language models and used the Good - Turing estimator [7] and Katz back-off smoothing [6] technique

# GOOD - TURING ESTIMATOR

- The general form of the estimator is:

$$P(X) = \frac{r^*}{N}$$

$$\text{where, } r^* = (r + 1) \cdot \frac{E(N_{r+1})}{E(N_r)}$$

- where:
  - $r^*$  is the adjusted number of occurrence;
  - $r$  is the number of occurrence of word  $X$  in the training corpus;
  - $N$  is the total number of words from the training corpus;
  - $N_r$  is the number of words which occurs exactly  $r$  times in the training corpus;
  - $E$  is an estimation function for  $N_r$ ;

$$\frac{E(n+1)}{E(n)} = \frac{n}{n+1} \cdot \left(1 - \frac{E(1)}{N}\right)$$

# BACK – OFF SMOOTHING

- All the probabilities for n-gram word sequences which has an occurrence number between 0 and a threshold value ( $K = 6$ ), will be smoothed using Good-Turing estimator to save probability mass for unseen word sequences [7].
- If a word sequence has zero occurrences in the trigram model, we try to estimate its probability using the inferior bigram model. If the occurrence is still zero for this inferior model we continue to back-off to the unigram model. Here, we always have the relative frequency of a word bigger than zero.

# LANGUAGE MODEL EVALUATION

- For the experiments we used perplexity to measure the quality of language models
- Perplexity is defined as:

$$PP = 2^{LP}$$

- We use the Romanian Constitution training corpus to estimate probabilities  $p^*$  and we use the logprob (LP) value:

$$LP = -\frac{1}{N} \log_2 p^*(w_1^n)$$

# TEXT CONDITIONING TOOL

- Our system performs the following basic conditionings:
  - segments text into sentences on the basis of punctuation, marking with <s> and </s> tags the beginning and the end of the sentences
  - puts only one sentence per line
  - stripe most punctuation symbols
  - convert numbers into words
  - converts the whole text to uppercase
  - deletes all empty lines
  - eliminates redundant white spaces

# ROMANIAN CONSTITUTION CORPUS

- Romanian Constitution corpus contains the entire Romanian constitution in raw text format
- Contains 9936 words
- The total number of distinct words in corpus (the vocabulary) is 1928 grouped in 718 sentences.
- 963 words has more than one appearance in the corpus.

# EXPERIMENTAL RESULTS

Table 1. Most frequent four-grams in Romanian Constitution corpus

| Word 1     | Word 2      | Word 3   | Word 4      | Number of appearance |
|------------|-------------|----------|-------------|----------------------|
| </S>       | <S>         | DREPTUL  | LA          | 27                   |
| DE         | LEGE        | </S>     | <S>         | 16                   |
| SE         | STABILESC   | PRIN     | LEGE        | 12                   |
| </S>       | <S>         | DREPTUL  | DE          | 11                   |
| PRIN       | LEGE        | ORGANICA | </S>        | 10                   |
| LEGE       | ORGANICA    | </S>     | <S>         | 10                   |
| </S>       | <S>         | CAMERA   | DEPUTATILOR | 9                    |
| CONDITIILE | LEGII       | </S>     | <S>         | 8                    |
| IN         | CONDITIILE  | LEGII    | </S>        | 8                    |
| CAMERA     | DEPUTATILOR | SI       | SENATUL     | 8                    |

# EXPERIMENTAL RESULTS

Table 2. The most probable 15 words from Romanian Constitution

| Word | Appearance | Word | Appearance | Word    | Appearance |
|------|------------|------|------------|---------|------------|
| </S> | 718        | A    | 195        | AL      | 71         |
| <S>  | 718        | LA   | 151        | DREPTUL | 70         |
| DE   | 470        | SE   | 128        | PRIN    | 69         |
| SI   | 405        | SAU  | 116        | PENTRU  | 68         |
| IN   | 287        | ESTE | 82         | CU      | 66         |

# EXPERIMENTAL RESULTS

- We generated a dictionary from the most probable 963 words from the corpus (the words with occurrence greater than 1) and we map all the other words into an unknown word class.
- We have generated the unigram, bigram, and trigram language models using Katz back-off smoothing

Table 3. Perplexity results for Romanian Constitution corpus using a 963 word dictionary.

| Model   | Perplexity |
|---------|------------|
| Unigram | 559.74     |
| Bigram  | 397.37     |
| Trigram | 419.52     |

# EXPERIMENTAL RESULTS

- We have made a second experiment, with a smaller dictionary, containing just the words with appearance greater than 2 (a number of 626 words).

Table 4. Perplexity results for Romanian Constitution corpus using a 626 word dictionary.

| <b>Model</b> | <b>Perplexity</b> |
|--------------|-------------------|
| Unigram      | 509.64            |
| Bigram       | 332.24            |
| Trigram      | 419.23            |

# CONCLUSIONS, FURTHER DEVELOPMENTS

- In both experiments the best results were achieved using the bigram model. The trigram model can't improve the results because there are insufficient data for training the model.
- We can see from the results that if the number of words increase, the perplexity of the model increase too, and the model has weaker quality
- The words which occurs only once does not improve the model quality, they rise perplexity and they should be eliminated from vocabulary (n-gram pruning) as we can see also in [3] – the results for Susanne corpus
- The n-gram model must be correlated with the corpus size.

# CONCLUSIONS, FURTHER DEVELOPMENTS

- We have to generate language models using state of the art LM tools for comparison
- We want to improve our text conditioning tool, to convert abbreviations, acronyms and other “non-standard words” (NSWs) to their spoken forms
- We want to create language models based on larger Romanian text corpus based on newspaper articles
- We want to implement more powerful smoothing techniques like Kneser-Ney smoothing [7]

# REFERENCES

1. BECCHETTI, C., RICOTTI, L. P., *Speech recognition. Theory and C++ implementations*, John Wiley & sons, 1999.
2. CHEN, S., GOODMAN J., *An Empirical Study of Smoothing Techniques for Language Modeling*, Harvard Computer Science Technical report TR-10-98, 1998.
3. DOMOKOS, J., TODEREAN, G., BUZA, O., *Statistical Language modeling on Susanne corpus*, IEEE International Conference COMMUNICATIONS 2008, Proceedings, pp. 69-72, 2008.
4. DOMOKOS J., TODEREAN G., *Continuous speech phoneme recognition using artificial neural networks*, Automation Computers Applied Mathematics, ISSN 1221-437X, Vol. 17 (2008) no. 1, pp. 5-11
5. HUANG, X., ACERO, A., Hon H., *Spoken Language Processing. A Guide to Theory, Algorithm & System Development*, Prentice Hall, 2001.
6. JELINEK, F., *Statistical Methods for speech recognition*, The MIT Press, 2001.
7. JURAFSKY, D., MARTIN, J. H., *Speech and language processing. An introduction to Natural language Processing. Computational Linguistics and Speech Recognition*, Prentice Hall, 2000.
8. MANNING, C., HEINRICH, S., *Foundations of statistical language processing*, The MIT Press, 1999.
9. ROSENFELD, R., *Two decades of statistical language modeling: where do we go from here?*, Proceedings of the IEEE, Volume 88, pp.: 1270 – 1278, 2000.
10. SCHWARM, S.; OSTENDORF, M., *Text normalization with varied data sources for conversational speech language modeling*, IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP '02, Volume 1, pp. 789 – 792, 2002.
11. YOUNG, S., EVERMANN, G., GALES, M., HAIN, T., KERSHAW, D., MOORE, G., ODELL, J., OLLASON, D., POVEY, D., VALTCHEV, V., WOODLAND, P., *The HTK Book*, Cambridge University Engineering Department, 2005.
12. PAUL, D., B., BAKER, J., M., *The design for the wall street journal-based CSR corpus*, Workshop on Speech and Natural Language, Proceedings, pp. 357 – 362, 1992.