CAPITALIZATION AND PUNCTUATION RESTORATION FOR ROMANIAN LANGUAGE

Alexandru CARANICA[1], Horia CUCU[2], Andi BUZO[3], Corneliu BURILEANU[4]

The text generated by an Automatic Speech Recognition system is usually characterized by low reading intelligibility. Capitalization, also known as truecasing, is the process of restoring case information to badly-cased or noncased text. Punctuation recovery or restoration is the process of inserting punctuation marks (at least periods and commas) in a punctuation-lacking text. In this paper, we present an integrated capitalization and punctuation restoration solution for a Romanian ASR system. The solution implements both tasks in a single system (framework) and uses statistical information from a set of tri-gram language models as a post-processing stage. The integrated system is evaluated in terms of precision, recall and f-measure.

Keywords: speech recognition, punctuation recovery, capitalization, language modeling

1. Introduction

The output of an Automatic Speech Recognition (ASR) system consists of raw text, often in lowercase format and without any punctuation information. The transcript is intended to be as close as possible to the speech content of the audio file [1]. This may be useful for a wide range of applications, such as database indexing and classification, where a machine uses this information in search related algorithms. For other tasks, where humans need to easily read and understand the text (e.g. subtitling, dictation and broadcast news transcription), capitalization and punctuation restoration greatly improves the readability of automatic speech transcripts. Apart from the insertion of punctuation marks and capitalization, enriching speech recognition covers other activities, such as detection and filtering of disfluencies, sentence segmentation, but these are not part of this study.

[1] PhD student, Speech and Dialogue Research Laboratory, University POLITEHNICA of Bucharest, Romania, e-mail: alexandru.caranica@yahoo.com
[2] Lect., Speech and Dialogue Research Laboratory, University POLITEHNICA of Bucharest, Romania
[3] Prof., Speech and Dialogue Research Laboratory, University POLITEHNICA of Bucharest, Romania.
[4] Prof., Speech and Dialogue Research Laboratory, University POLITEHNICA of Bucharest, Romania

In ASR systems, researchers in the field tried to use prosodic information, disfluencies and overlapping speech, to predict punctuation, and later they have supplemented these techniques with language models [2].

According to A. Gravano, the approaches based on acoustic and prosodic information significantly outperform the methods based purely on *n*-gram models [3]. This is concluded after multiple experiments with data-driven techniques for annotating transcribed speech with sentence boundaries [4][5], and with sentence boundaries and other punctuation symbols, predominantly commas and question marks [6][7][8]. At the same time, digitized text data is growing exponentially in volume, and the availability of massive amounts of written data, coupled with progress in computational power and storage capacity ("cloud model"), asks the question of the extent to which text-based models may be improved when increasing both the training data size and the *n*-gram order [3]. Such text data is often produced automatically (ex. via speech recognition or optical character recognition) or in a hurry or unstructured manner (like instant messaging or web user forum data). Hence this data contains noise and needs to be first cleaned and processed in order to obtain any usable data for training, corpus creation, etc.

In this paper we describe a set of experiments regarding language model generation, training and evaluation in the context of capitalization and punctuation recovery for the Romanian language. Although the methodology is not new, to the best of our knowledge this is the first such system developed for the Romanian language and these are the first re-capitalization and punctuation restoration results reported for this language. The *n*-gram language models are trained with data varying from 44 million to 290 million words of written Romanian text. The training and evaluation data consists of broadcast transcriptions and online news and was previously collected by our research group. The restoration system was eventually integrated into our large-vocabulary automatic speech recognition for Romanian.

The remainder of this paper is structured as follows. Section 2 presents previous and different approaches to capitalization and punctuation restoration in the context of speech processing, generally for English (at the time this paper was written, we could not find a relevant study for Romanian). Section 3 and 4 presents the methodology for capitalization and punctuation restoration in detail, along with information about the training and evaluation data used in this paper. In Section 5 we present the experimental results of our tests with the proposed methodology and discuss the results, while Section 6 lists conclusions and future work directions.

2. Related work

Spoken language is similar to written text in many aspects, but differs due to the way these communication methods are produced. Current ASR systems are evaluated based on the WER (Word Error Rate), which does not take into account the detection of structural information available in written texts. As a result, case and punctuation restoration was a relatively unexplored field until this decade [2].

One of the first systems to use a simple hidden Markov model with trigram probabilities to model the comma and restoration problem was "cyberpunc", a lightweight method for automatic insertion of intra-sentence punctuation into text [9]. It restored the punctuation of 54% of the sentences correctly. Further work was done in [10] using syntactic information. This paper improves on previous study to achieve an accuracy of 58% for comma restoration. In both of these studies, sentence boundaries are assumed to be given at the input of the processing system. Because the above mentioned methods deal with punctuation restoration at the sentence level, this simplifies the task significantly, as the sentence boundaries are needed as a constraint, resulting in systems that are unable to process large quantities of raw unprocessed ASR text. Regarding case information, Lita [11] proposed a language model-based case restoration method, and the truecaser agreement with the original reference text is about 98%. The high precision reported in the quoted paper can be used as an indicator that the case restoration task is simpler when compared to the punctuation restoration task.

Regarding punctuation marks, a large number can be considered for ASR output texts, including: comma, period or full stop, exclamation mark, question mark, semicolon etc. However, most of these marks rarely occur and are quite difficult to insert or evaluate. Therefore, most of the available studies focus either on full stop or on full stop and comma, which have higher corpus frequencies [14]. A number of recent studies also consider the question mark [3], and even fewer consider other punctuation marks, such as exclamation marks.

A more recent study by A. Gravano (2009) [3] is of particular interest, not only because it uses *n*-gram language models, but also because of the large amount of training data, from 58 million to 55 billion tokens. He concludes that a) increasing the *n*-gram order does not significantly improve capitalization results and b) increasing the size of the training data improves both precision and recall for capitalization. These conclusions, combined with the adaptation of the training data set to domain specific data [12], were taken into account in our development of the restoration methodology for capitalization and punctuation. Gravano's study obtained a mean precision and recall of ~81% / 77% for capitalization, ~46% / 42% for comma and ~56% / 48% for period, for broadcast news reference transcript.

The results from the above mentioned study [3] served as a baseline benchmark of our system. We also note that all statistical capitalization and punctuation restoration systems presented in previous work utilize large amount of domain specific text corpora for training. To the best of our knowledge, these are the first re-capitalization and punctuation restoration results reported for Romanian language.

3. Methodology

Much of the prior research on punctuation restoration using *n*-gram models has been based largely on human transcriptions of speech, and so it has focused on retrieving / using textual information to train the language model. A language model (LM) describes possible word sequences, for the purpose of speech recognition and other language technologies. Statistical language modeling (SLM) attempts to capture regularities of spoken language in order to improve the performance of various natural language applications [17]. We use SLM to estimate the probability distribution of various linguistic units (such as word tokens) and sequences of linguistic units. The language model decomposes the probability of a sentence (*s*) into a product of conditional probabilities:

$$\Pr(s) = \Pr(w_1 ... w_n) = \prod_{i=1}^{n} \Pr(w_i \mid h_i) \tag{1}$$

where w_i is the *i*-th word in the sentence, and $h_i = \{w_1, w_2, ... w_{i-1}\}$ is called a history and, in this case, is a string of *i* tokens. An *n*-gram reduces the dimension of the estimation problem by modeling the language as a Markov source of order *n*-1:

$$\Pr(w_i \mid h_i) \approx P(w_{i-n+1}, ..., w_{i-1}) \tag{2}$$

where the approximation reflects a Markov assumption that only the most recent $n-1$ tokens are relevant when predicting the next token. We train with a value of $n=3$, as trigrams are a common choice with large training corpora (millions of tokens) [17], and [3] shows that increasing the *n*-gram order does not help as much as increasing the training data set. A language model quality is measured by its effect on the specific language application for which it was designed, namely by improving the word error rate of that application. This is influenced by the quality of the *n*-gram language model. For under-resourced languages, like Romanian, significant efforts were made by the "SpeeD" group in recent years to increase the quality and size of the recorded speech corpus in Romanian, collect and pre-process new information from the Internet, in order to obtain better performing *n-gram* models after the training process [15].

Figure 1 presents an overview of the system architecture and illustrates a) the role of the capitalization and punctuation restoration module, as a post-processing module for the transcripts resulted out of an automatic speech recognition process and b) the training processes which need to be employed to generate the *n*-gram language model.

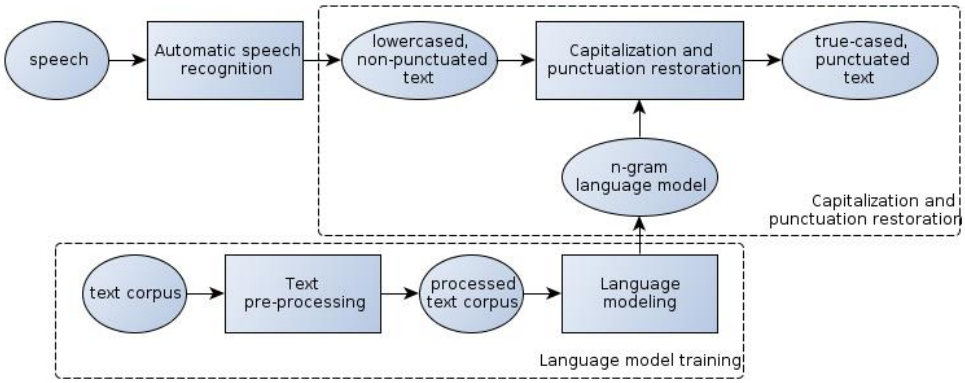

Fig. 1. Capitalization and punctuation restoration module

The algorithm used in the capitalization and punctuation restoration module processes the input text line by line. The words on each line of input text are processed one by one, from left to right. The n^{th} word on the input line is appended to the existing sequences of *n-1* words for that particular line. The word is appended to each existing sequence in all its possible capitalization forms: lowercased (e.g. popa), capitalized (Popa) and all-caps (POPA) and followed by all the possible punctuation marks took into account by our study: no punctuation mark, comma, period. For example, suppose that one of the existing sequences of words for the current line of text is "M-am întâlnit cu" and that the next input word is "popa". The current sequence of three words is expanded into the following 6 sequences of four words:

1) M-am întâlnit cu popa
2) M-am întâlnit cu Popa
3) M-am întâlnit cu popa <period>
4) M-am întâlnit cu Popa <period>
5) M-am întâlnit cu popa <comma>
6) M-am întâlnit cu Popa <comma>

In this example the word form "POPA" is not a valid word in Romanian so it is not taken into account when forming the sequences of four words. After the new sequences are generated, their probability is scored using the *n*-gram language model. Whenever the number of word-sequences of a certain length exceeds a given threshold, the list of sequences is pruned (the sequences with the lowest probabilities are discarded). After all the words on the input line were

processed, the word sequence with the highest probability is sent to the output. This algorithm is inspired from a similar implementation within the CMU Sphinx ASR toolkit [16].

The success of the above algorithm is directly influenced by the quality of the *n-gram* language model. The language model needs to model as well as possible the probabilities for the words and punctuation tokens. The key features in the language modeling part are the *n*-gram order and the size, quality and adequacy of the training text corpus. The methodology we propose for text pre-processing and conditioning operations (for the Romanian language) are the following:

1) Diacritics and hyphens uniformization. Generally in Romanian texts there are several character codes (incorrectly) used for the same diacritical characters (e.g. ã, ă; ş, ș; etc.) and several hyphen character codes (wrongly) used to form compound words. For a correct computation of statistics for diacritical and hyphen words and word sequences, these characters have to be used in a consistent manner.
2) Diacritics restoration. Most Romanian texts are incorrectly written without diacritics. For the same reason (correct computation of statistics for diacritical words), these texts have to be conditioned: diacritics to be restored.
3) Replace punctuation marks with a corresponding token. As opposed to the ASR approach, in the case of punctuation restoration, the language model has to model the statistics of punctuation tokens as well. Because in this study we approached the restoration of commas and periods, all the other punctuation marks were mapped to one of these tokens, as described in Table 1.

Table 1:

The correspondence between the punctuation marks and tokens in the LM

Punctuation mark	Token
, ( ) --	<COMMA>
: ; ! . ? ...	<PERIOD>

4. Evaluation setup

For a thorough evaluation of the proposed restoration methodology we used two large Romanian text corpora previously collected by the "SpeeD" research group over the Internet. As described in Table 2, for the training process we used the two corpora separately and together to create three different language models. Table 2 also illustrates the number of tokens in each corpus and the average number of tokens per paragraph.

The average number of tokens per paragraph is especially important for evaluation, because the algorithm automatically inserts period at the end of every processed line. In the case of short paragraphs (few words per line) this might artificially increase the punctuation score. We do not see this as a problem for our usage scenario because the average number of tokens in each paragraph is quite large (over 45).

Table 2:

The language models and training corpora

Language model	Training corpora	Tokens	Token/Paragraph
TalkshowsLM	talkshows	45M	45
NewsLM	news	243M	67
MergedLM	talkshows + news	288M	63

For the evaluation process we used two held-out sets of data from the two Romanian corpora. The evaluation corpora contain 100k paragraphs each, with approximately 4M word tokens each. The evaluation corpora were pre-processed exactly as the training corpora (see Section 3) to become similar to the real output of a speech recognition system. In addition to this pre-processing operation, the evaluation data goes through an extra process to remove (lowercase) capitalization and eliminate all punctuation marks.

As performance figures, we used the standard criteria for evaluation of punctuation restoration and capitalization: precision, recall and f-measure.

$$precision = \frac{C}{C + I} \cdot 100 \tag{3}$$

$$recall = \frac{C}{C + D} \cdot 100 \tag{4}$$

$$f - measure = \frac{precision \cdot recall}{precision + recall} \cdot 100 \tag{5}$$

In these equations, C represents the number of correct tokens, I is the number of insertion errors and D is the number of deletion errors or missing tokens.

For capitalization, the correctly capitalized words are counted as correct (C), the words that are capitalized in the reference, but not in the hypothesis are counted as deletions errors (D) and the words that are wrongly capitalized in the hypothesis and are not capitalized in the reference are counted insertion errors (I). Table 3 shows an example performance measures for both capitalization and punctuation restoration.

Table 3:

Example of evaluation procedure for punctuation restoration and capitalization

REF: Acesta este un exemplu , de calcul . HYP: Acesta , este un exemplu de calcul . I D C
REF: Acesta este un Exemplu de CALCUL HYP : Acesta Este un exemplu de CALCUL C I C D C C

5. Evaluation results and discussion

Tables 4, 5 and 6 contain the results. As stated in previous section, the two test corpora were evaluated against all three trained language models.

Table 4:

Precision, Recall and F-measure for the talkshows evaluation corpus.

Language Model	**Capitalization**		
	Precision	**Recall**	**F-measure**
TalkshowsLM	80%	69%	74%
MergedLM	76%	72%	73%
NewsLM	72%	71%	71%
Language Model	**Comma**		
	Precision	**Recall**	**F-measure**
TalkshowsLM	55%	46%	50%
MergedLM	56%	43%	48%
NewsLM	50%	38%	43%
Language Model	**Period**		
	Precision	**Recall**	**F-measure**
TalkshowsLM	68%	50%	57%
MergedLM	64%	54%	58%
NewsLM	59%	54%	56%

Table 5:

Precision, Recall and F-measure for the news evaluation corpus.

Language Model	**Capitalization**		
	Precision	**Recall**	**F-measure**
TalkshowsLM	80%	46%	58%
MergedLM	80%	66%	72%
NewsLM	80%	66%	72%
Language Model	**Comma**		
	Precision	**Recall**	**F-measure**
TalkshowsLM	49%	33%	39%
MergedLM	64%	48%	54%
NewsLM	67%	54%	59%
Language	**Period**		

Model	Precision	Recall	F-measure
TalkshowsLM	68%	34%	45%
MergedLM	68%	53%	59%
NewsLM	67%	54%	59%

Accuracy is measured for words only (capitalized and non-capitalized), excluding punctuation marks, in the entire hypothesis output:

$$accuracy = \frac{C}{T} \cdot 100 \tag{6}$$

where C is the number of correct words and T is the total number of words.

Table 6:

Word Accuracy for both corpuses

Language Model	Word Accuracy for talkshows test data	Word Accuracy for news test data
TalkshowsLM	94%	89%
MergedLM	93%	92%
NewsLM	93%	92%

Figures 2 and 3 summarize the results and show a visual representation of all the corresponding values in the above tables.

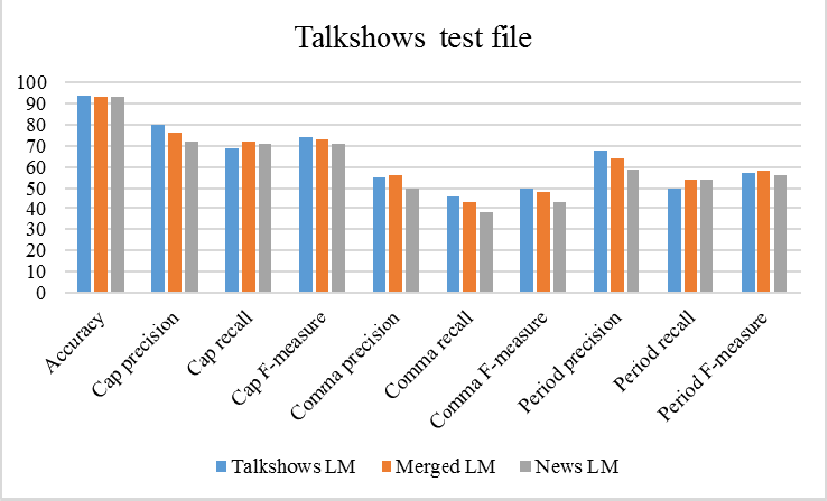

Fig. 2: Visual representation of talkshows test results

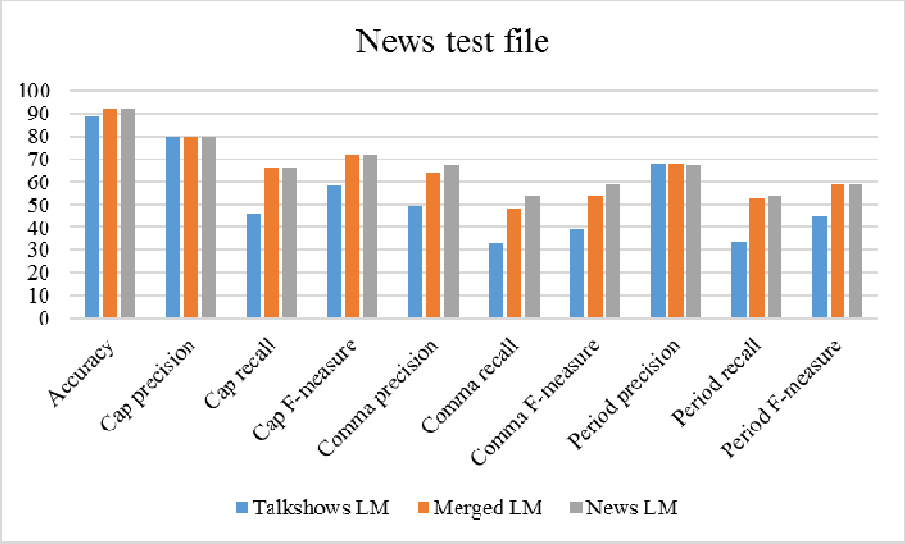

Fig. 3: Visual representation of news test results

To assess the impact on metrics, we prepared three data sets with a varying numbers of tokens: 45M, 243M and 288M, as shown in section 4. We trained a *tri*-gram language model for each data set, considering two punctuation tokens. As the visual representation figures show, increasing the corpus size has a positive impact on most of the performance metrics, and can also state that corpus domain has an impact on results. We can conclude that further increasing the size of the training data, coupled with a more complex LM, from the same domain as the evaluation corpora, will presumably increase performance. Table 7 further illustrates the impact on readability of a paragraph from Romanian news, where the output from a transcribed news paragraph is compared against the hypothetical and ideal output.

Table 7:

Comparison of ASR output with / without capitalization and punctuation restoration, on a Romanian news paragraph

Raw ASR output
iată ce spun telespectatorii noştri pe facebook în continuare îi rog să ne trimită propuneri pentru guvernul ponta
ASR output with capitalization and punctuation restoration
Iată ce spun telespectatorii noştri pe Facebook, în continuare îi rog să ne trimită propuneri pentru guvernul Ponta.
Ideal ASR output
Iată ce spun telespectatorii noştri, pe Facebook. În continuare, îi rog să ne trimită propuneri pentru Guvernul Ponta.

6. Conclusion and future work

This paper presents an approach to restore punctuation and capitalization for text in Romanian language, using text-based tri-gram language models. Overall, our tests show a precision of 76-80% for capitalization restoration, 54-60% for comma and 64-68% for period recovery. Restoration of diacritics in Romanian is necessary for all future test corpora, if missing. Otherwise, the algorithm will threat capitalized and uncapitalized words as different tokens and fail to restore capitalization for text without diacritics. Furthermore, our results suggest that test files from the same corpora domain offer better results, by a small margin. This margin can be reduced by using a larger training corpus in order to obtain better performing language models.

In conclusion, the punctuation and capitalization restoration tasks greatly improve the intelligibility of the ASR output (as shown in table 7), even though its accuracy and precision are not 100%. The availability of large unstructured data over the internet, that can be downloaded and processed, makes this LM-based restoration principle a feasible method for this task.

Further work will focus on improving the language models, extending the study on other punctuation marks and on including more complex models based on acoustic/prosodic features from the audio signal.

Acknowledgements

The work has been funded by the Sectoral Operational Programme Human Resources Development 2007-2013 of the Ministry of European Funds through the Financial Agreements POSDRU /159 /1.5/ S/ 132395, and POSDRU /159 /1.5 /S/134398.

R E F E R E N C E S

[1] *A. Buzo, H. Cucu, L. Petrică, D. Burileanu and C. Burileanu*, "An Automatic Speech Recognition Solution with Speaker Identification Support", Proceedings of COMM, pp. 119-122, 2014.

[2] *T. Baldwin, M. P. A. K. Joseph*, "Restoring Punctuation and Casing in English Text", Springer (LNCS), 2009.

[3] *A. Gravano, M. Jansche, M. Bacchiani*, "Restoring punctuation and capitalization in transcribed speech", in ICASSP, 978-1-4244-2354-5, 2009.

[4] *E. Shriberg, A. Stolcke, D. Hakkani-Tur, and G. Tur*, "Prosodybased automatic segmentation of speech into sentences and topics," Speech Comm., vol. 32(1-2), pp. 127–154, 2000.

[5] *Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, M. Harper*, "Enriching speech recognition with automatic detection of sentence boundaries and disfluencies," IEEE Trans. Audio Speech Lang. Process., vol. 14(5), pp. 1526–1540, 2006.

[6] *E.W. Brown, A.R. Coden*, "Capitalization recovery for text", IR Techniques for Speech

Applications, 2002.
[7] *H. Christensen, Y. Gotoh, S. Renals*, "Punctuation annotation using statistical prosody models", in ISCA Workshop on Prosody in Speech Recognition and Understanding, 2001.
[8] *B. Favre, R. Grishman, D. Hillard, H. Ji, D. Hakkani-Tur, M. Ostendorf*, "Punctuating Speech for Information Extraction", in ICASSP, 2008
[9] *D. Beeferman, A. Berger, and J. Lafferty*. "Cyberpunc: A lightweight punctuation annotation system for speech", in Proceedings of 1998 IEEE ICASSP'98, Seattle, USA, 1998.
[10] *S. M. Shieber, X. Tao*. "Comma restoration using constituency information", in Proceedings of the 3rd International Conference on Human Language Technology Research, Edmonton, Canada, 2003.
[11] *L. V. Lita, A. Ittycheriah, S. Roukos, N. Kambhatla*, "tRuEcasIng", in ACL, Japan 2003.
[12] *C. Chelba and A. Acero*, "Adaptation of maximum entropy capitalizer: Little data can help a lot", Computer Speech & Language, 20(4):382–399, 2006
[13] *F. Batista, H. Moniz, I. Trancoso, N. Mamede*, "Bilingual experiments on automatic recovery of capitalization and punctuation of automatic speech transcripts", in Audio, Speech, and Language Processing, IEEE, 2012
[14] *F. Batista, D. Caseiro, N. Mamede, I. Trancoso*, "Recovering Capitalization and Punctuation Marks for Automatic Speech Recognition: Case Study for Portuguese Broadcast News, Speech", in Speech Communication, 2008.
[15] *H. Cucu, A. Buzo, L. Petrică, D. Burileanu, C. Burileanu*, "Recent Improvements of the SpeeD Romanian LVCSR System", in Proceedings of COMM, 2014.
[16] ***http://cmusphinx.sourceforge.net/wiki/postpframework, last accessed July 2014.
[17] *R. Rosenfeld*, "Two decades of statistical language modeling: where do we go from here?", Proceedings of the IEEE, 2000.