

# The CMU Statistical Language Modeling Toolkit and its use in the 1994 ARPA CSR Evaluation

*Ronald Rosenfeld*

School of Computer Science  
Carnegie Mellon University  
Pittsburgh, Pennsylvania 15213

## ABSTRACT

The Carnegie Mellon Statistical Language Modeling (CMU SLM) Toolkit is a set of Unix software tools designed to facilitate language modeling work in the research community. The package, including source code, is freely available for research purposes. As of December 1994, the toolkit is in active use by 23 research groups in 8 countries. It was recently used to process the 2.5 GB NAB corpus for the ARPA CSR community. In this paper, I first discuss the design principles and features of the toolkit. Then, I describe the composition of the NAB corpus, and report on the ngram statistics, standard vocabulary and language models created using the SLM tools.

## 1. OVERVIEW OF THE CMU SLM TOOLKIT

### 1.1. Introduction

The Carnegie Mellon University Statistical Language Modeling (CMU SLM) Toolkit is a set of Unix software tools designed to facilitate language modeling work in the research community.

Some of the tools are used to process general textual data into:

- word frequency lists and vocabularies
- word bigram and trigram counts
- vocabulary-specific word bigram and trigram counts
- bigram- and trigram-related statistics
- various Backoff [Katz 87] bigram and trigram language models

Other tools use the resulted language models to compute:

- perplexity
- Out-Of-Vocabulary (OOV) rate
- bigram and trigram hit ratios
- distribution of Backoff cases
- annotation of test data with language scores

The main motivation behind the CMU SLM Toolkit is to facilitate language modeling research. Conventional language modeling technology can be learned quite quickly by reading a good tutorial (see for example [Jelinek 89]). But for a research group beginning work on a human language project for which statistical language modeling is useful, creating the necessary programs may be tedious, time consuming, and prone to error. The toolkit is meant to obviate the need for this “reinvention of the wheel”, and to allow such groups

to devote their resources to other aspects of their system, or to further improvements in language modeling technology. To facilitate this last goal, the tools are designed as modular building blocks that can be manipulated and modified, rather than as frozen recipes for producing a final model.

Another intended use for the tools is to allow different sites to meaningfully compare their baseline LM technology. Although the theoretical concepts behind backoff [Katz 87], linear interpolation [Jelinek & Mercer 80], Good-Turing discounting [Good 53] and the like are quite clear to everyone in the field, there are still many implementation details that are left unspecified in the literature. Such details can make a non-negligible difference in the final product, making cross-implementation comparisons harder to interpret. With the easy availability of the SLM Toolkit, its products can be used as a benchmark against which other implementations can be compared.

The CMU SLM Toolkit, including source code, is freely available by anonymous ftp from Carnegie Mellon’s ftp server. See appendix A for instructions.

### 1.2. Design Principles

The approach I took in designing the toolkit is that of simple step-by-step manipulation of data streams. Thus whenever possible I avoided large, complicated programs. Instead, I favored a collection of short simple programs that can be piped together to achieve the same functionality. This allows for ‘splicing-in’ of new steps and for surgical modification of the data, both being crucial for language modeling research. When the tools are piped together, no extra disk I/O is incurred and the additional overhead is minimal.

Thus most of the SLM tools are simple filters that convert a data stream into a slightly different format. A typical low-level tool is named ‘abc2def’, meaning that it processes a .abc stream into a .def stream. A typical higher-level tool is named ‘abcTOxyz’, meaning that it uses a combination of lower-level tools to process a .abc stream into a .xyz stream. Exact definitions for all data stream formats are provided, so the user can create their own high-level tools by concatenating existing tools together. A few sample definitions are shown in table 1. When data is stored in a file, the data’s format is implied by the filename’s extension.

### 1.3. Context Cues

The SLM tools view language data as a stream of words, possibly interspersed with “context cues”. These are markers that are not a part of the application vocabulary, but which provide some context for the text around them, and are therefore potentially useful for language modeling. Currently, the following context cues are defined (in

<code>.text</code>	Text, words separated by whitespace.
	<ul style="list-style-type: none"> <li>• May contain any printable character.</li> <li>• Case is important.</li> <li>• Beginning-of-sentence must be designated with '<code>&lt;s&gt;</code>'.</li> <li>• End-of-sentence must be designated with '<code>&lt;/s&gt;</code>'.</li> <li>• Beginning-of-paragraph may be included; if so, it must be designated with '<code>&lt;p&gt;</code>'.</li> <li>• Beginning-of-article may be included; if so, it must be designated with '<code>&lt;art&gt;</code>'.</li> </ul>
<code>.wunic</code>	Word UNIGram Counts.
	<ul style="list-style-type: none"> <li>• Every line starts with: <code>&lt;word&gt; &lt;count&gt;</code></li> <li>• Rest of line is ignored.</li> <li>• May include context-cues.</li> <li>• Sorted lexicographically by the first field.</li> </ul>
<code>.wtric</code>	Word TRIgram Counts.
	<ul style="list-style-type: none"> <li>• Every line is of the form: <code>&lt;word1&gt; &lt;word2&gt; &lt;word3&gt; &lt;count&gt;</code></li> <li>• Sorted lexicographically by the first three fields.</li> <li>• a special case of a <code>.vtric</code> stream.</li> </ul>
<code>.bbo</code>	Binary Back-Off Ngram language model.
	<ul style="list-style-type: none"> <li>• Binary, machine-byteorder independent format.</li> <li>• Could be either a bigram or a trigram.</li> </ul>

Table 1: Sample definitions of data stream formats used by the CMU SLM Toolkit. Such streams are both inputs and outputs of the tools.

agreement with the SGML markers used in LDC and ARPA/CSR data):

- `<art>`: beginning-of-article/document marker
- `<p>`: beginning-of-paragraph marker
- `<s>`: beginning-of-sentence marker

Of these three markers, only `<s>` is required in the LM vocabulary. The others are optional. If the original data contains paragraph and document boundary markers, one may map them into the above symbols. The SLM tools treat these markers in a special way. During training, they will be used as part of the context (of, say, bigrams and trigrams) but will not be modeled themselves. When a language model built by the SLM tools is used, it will identify these symbols in the test data and will interpret them appropriately. For example, in  $N$ -gram models, inclusion of '`<p>`' allows one to model the first word of a paragraph separately from the first word of a sentence. More importantly, caches and other adaptive models can use the context cues as implicit instructions to perform incremental adaptation, flush their internal context, etc.

'`</s>`' is a related but very different special symbol. It stands for 'end-of-sentence', and must be part of the application vocabulary. It is modeled like any other word in the vocabulary. In particular, the probability of an isolated sentence  $S \stackrel{\text{def}}{=} (W_1, W_2, \dots, W_n)$  is

typically computed as:

$$\Pr(S) = \prod_{i=1}^{n+1} \Pr(W_i | W_1, W_2, \dots, W_{i-1})$$

where  $W_{n+1} \stackrel{\text{def}}{=} \text{'</s>'}$ .

## 1.4. Vocabularies and OOVs

Much of the processing done by the SLM tools is done with no restriction on the words present in the data stream (other than the special symbols discussed above). But conventional language models are defined relative to a particular vocabulary. As a step towards creating such models, all words that are outside the given vocabulary (Out-Of-Vocabulary words, or OOVs) are mapped into a single symbol, called the 'UNK' ("unknown") symbol.

The SLM tools support the construction of 'closed-vocabulary' language models, namely models which do not expect to see in their input any words outside their predefined vocabulary. More often, an "open-vocabulary" model may be more appropriate, where the possibility of OOVs is acknowledged and modeled in some way. The actual behavior of OOVs in the data depends greatly on whether that data was used in constructing the vocabulary. Consequently, two types of open-vocabulary models are supported by the SLM tools.

## 1.5. Storage Space

No assumptions are made by the tools about the amount of data to be processed. The only limiting factors are CPU and disk space. When disk space is limited, there are tools to manipulate the data in chunks, reduce each output to a compact form, then combine the outputs.

When disk space is limited, one may wish to store both intermediate and final files in a compressed form. The SLM tools support transparent compression/decompression of input and output, cued by the use of an appropriate filename extension.

## 1.6. Future Plans

Version 2.0, scheduled for mid 1995, includes faster preprocessing, more compact memory representation, some general purpose LM utilities, and support for linear interpolation and weight optimization.

# 2. PROCESSING THE 1994 NAB TEXT CORPUS WITH THE SLM TOOLKIT

## 2.1. The 1994 NAB Text Corpus

The 1994 North American Business (NAB) text corpus is a collection of textual data from a variety of news sources, spanning the years 1987–1994. It was obtained, processed and published by the Linguistic Data Consortium (LDC) based on desiderata provided by the ARPA CSR community. The data consists of text from the following North American Business news sources<sup>1</sup>:

**AP**: Associated Press wire feed (1988-1990).

**SJM**: San Jose Mercury News (1991).

<sup>1</sup>Data from several other sources, such as Reuters and The New York Times, could not be secured in time for this year's deadline, but is planned for the successor corpus.

**WSJ:** The Wall Street Journal (1987-1992).

**DJIS:** The Dow Jones Information Service (a superset of the WSJ source; 1992-1994).

The data was first cleaned, segmented and conditioned for linguistic research by Dave Graff at LDC. It was then further conditioned for use in speech recognition, using Doug Paul's text conditioning tools [Paul & Baker 92], which were improved and updated to better handle this corpus. This phase of the conditioning was done in parts by Dave Graff at LDC and by Nina Yuan at BBN. Another part of the data was copied over from the predecessor corpus (WSJ0), which was originally conditioned by Doug Paul at MIT-LL.

The output of the last stage, about 2.5GB of text, was sent to the author at Carnegie Mellon, for processing into vocabularies, ngram counts and language models, using the SLM Toolkit.

Since the corpus was conditioned at multiple sites and under severe time pressure, some conditioning errors and processing glitches were only discovered after the data arrived at Carnegie Mellon. Returning the data to LDC for reprocessing was not feasible due to the time constraints. Instead, I used the following improvised fixes:

1. Unprintable characters were mapped to '#'.
2. Lines containing gross SGML format violation were discarded.

These fixes were sent to LDC and were subsequently incorporated into the version of the corpus which LDC published on two cdroms (22-1.1 and 22-2.1). Many conditioning errors still remained, effecting an estimated 0.2%–0.5% of the tokens. This was judged tolerable, at least for this year. A repaired version of the corpus is scheduled to be republished by LDC in the near future.

## 2.2. Postprocessing at Carnegie Mellon

The NAB corpus as provided to Carnegie Mellon was in a 'VP' (Verbalized Punctuation) format, meaning that all punctuation marks were represented by special tokens (e.g. ',COMMA' for the ',' character). For the purpose of creating the vocabularies, count files and language models, the data was first transformed into an 'SVP1' format (Some Verbalized Punctuation, version 1), defines as follows:

Starting from the VP format, map:

@AT-SIGN	⇒	AT
&AMPERSAND	⇒	AND
+PLUS	⇒	PLUS
=EQUALS	⇒	EQUALS
%PERCENT	⇒	PERCENT
/SLASH	⇒	SLASH
.POINT	⇒	POINT
"DOUBLE-QUOTE	⇒	QUOTE (every 5th occurrence only)

and discard all other VP words.

The SVP1 format was designed to mimic as closely as reasonably possible the way punctuations are actually pronounced by subjects.

## 2.3. N-gram Statistics

After conversion to SVP1, the corpus was tallied using the SLM Toolkit. It consisted of 227 million token words in some 10 million sentences. The word and token counts (not including ⟨s⟩ or ⟨/s⟩),

broken down by component, or "chunk", were as follows:

chunk	# of tokens	# of words
WSJ87	17.3M	115K
WSJ88	14.5M	109K
WSJ89	5.5M	72K
WSJ90	9.6M	93K
WSJ91	18.4M	126K
WSJ92	4.3M	66K
DJIS92	15.6M	118K
DJIS93	20.5M	135K
DJIS94	5.2M	72K
AP88	33.1M	173K
AP89	39.6M	191K
AP90	32.7M	173K
SJM91	10.6M	103K
total	227M	476K

Of the 476K different words in the corpus, 182K occurred only once, and 65K occurred twice. Many of these low-count "words" were typing errors or unusually spelled foreign names.

Next, using the SLM Toolkit, word trigram counts were created for each chunk, and then merged together. There were 69 million different word trigrams, and the compressed trigram counts file occupied 435MB. From that file, trigram and bigram count statistics were computed, as follows:

this many trigrams	this many bigram	occurred this many times
51,178K	9,900K	1
8,442K	2,363K	2
3,051K	1,037K	3
1,579K	600K	4
962K	394K	5
4,045K	2,276K	>5
69,258K	16,570K	>0

## 2.4. The official 1994 CSR 20K Vocabulary

The official 1994 CSR vocabulary was designed for use in the vocabulary-restricted part of the ARPA CSR evaluation. It was constructed using the most frequent 20,000 words in the NAB corpus. The least frequent words in this vocabulary occurred 367 times in the corpus. Several dozen "junk" words, caused by the conditioning errors discussed above, were excluded. At least one such "word" (IFYOU'RE) was not detected in time to be removed.

The Out-Of-Vocabulary (OOV) rate of the corpus with regard to this vocabulary is 3.1% (~ 7M OOV tokens out of 227M). The OOV rate of the LM development set (a.k.a. "set-aside" text) is 3.05% for the DJIS portion and 2.8% for the rest.

Using the SLM Toolkit, the corpus' word trigram counts were mapped down to vocabulary-specific trigram counts: OOV words were mapped to the "UNK" symbol, and identical trigrams were merged. This resulted in 58 million trigrams (41M of which occurred once, 7.7M twice and 3M three times) and some 10 million bigrams (5M of which occurred once).

## 2.5. The official 1994 CSR Language Models

The official 1994 CSR language models (LMs) were designed for use in the vocabulary- and LM-restricted part of the ARPA CSR evaluation. Both a trigram and a bigram backoff models [Katz 87] were created. Both models were based on the official 20K vocabulary. In the trigram model, singleton bigrams and up to tripleton trigrams were excluded, leaving 6.7M trigrams and 5M bigrams, and resulting in a 98MB compressed trigram LM file. In the bigram model, singleton bigrams were excluded, leaving 5M bigrams and resulting in a 35MB compressed bigram LM file. These specific cutoffs were chosen to comply with the prevailing sentiment in the community, according to which the LMs should be larger than last year's LMs by no more than a factor of 2–3.

Perplexity of the LM development data (excluding prediction of the UNK symbol) is presented below. The analogous figures for the official 1993 LMs are included for comparison. Note that the 1993 LMs are based on a small subset of the NAB corpus (WSJ87-89, 39M tokens), have about half as many bigrams and trigrams, and use a different 20KW vocabulary.

LM	94 LM dev set (DJIS)	94 LM dev set (others)
1994 trigram	117	120
1993 trigram	144	153
1994 bigram	191	189
1993 bigram	204	211

## 2.6. Publication

The word frequency list, word trigram counts, vocabulary, vocabulary-specific trigram counts, language models and related statistics were all shipped to LDC, together with the SLM Toolkit. LDC subsequently published this data on two cdroms (22-3.1 and 22-4.1).

## Acknowledgements

I am grateful to Raj Reddy for his support of the SLM Toolkit project; to Francis Kubala for his help in making the NAB processing a success; to Doug Paul for much appreciated advice; to Mitch Weintraub and Doug Paul for timely testing of the NAB language model; to Bob Weide for invaluable logistical support in shipping and handling the NAB data, and to Ralf Brown, Jean-Luc Gauvain, Xiaoqiang Luo, Alexander Rudnicky and Bernhard Suhm for much appreciated feedback on the beta version of the Toolkit.

## References

- [Good 53] Good, I. J., “The Population Frequencies of Species and the Estimation of Population Parameters”, *Biometrika*, Volume 40, parts 3,4, pages 237–264, 1953.
- [Jelinek & Mercer 80] Jelinek, F. and Mercer, R., “Interpolated Estimation of Markov Source Parameters from Sparse Data”, In *Pattern Recognition in Practice*, E. S. Gelsema and L. N. Kanal (editors), pages 381–402. North Holland, Amsterdam, 1980.
- [Jelinek 89] Jelinek, F., “Self Organized Language Modeling for Speech Recognition”, in “*Readings in Speech Recognition*”, Alex Waibel and Kai-Fu Lee (Eds.), Morgan Kaufmann, 1989.
- [Katz 87] Katz, S. M., “Estimation of Probabilities from Sparse Data for the Language Model Component of a Speech Recognizer”, in “*IEEE Transactions on Acoustics, Speech and Signal Processing*”, volume ASSP-35, pages 400–401, March 1987.

[Paul & Baker 92] Paul, D. B. and Baker, J. M., “The Design for the Wall Street Journal-based CSR Corpus”, in *Proceedings of the DARPA SLS Workshop*, February 1992.

## APPENDIX A: ACQUIRING THE CMU SLM TOOLKIT

To get the CMU SLM Toolkit, follow these instructions:

1. `ftp ftp.cs.cmu.edu`
2. Login as anonymous
3. Provide your `userid@site` as password
4. `cd project/fgdata`
5. `binary`
6. `get CMU_SLM_Toolkit_V1.0_release.tarlog`
7. `get CMU_SLM_Toolkit_V1.0_release.tar.Z`  
(1.3MB)

To subscribe to the SLM Toolkit mailing list, send mail to: `cmu-slm-toolkit-request@cs.cmu.edu`