An Evaluation of Adopting Language Model as the Checker of Preposition Usage

Shih-Hung Wu, Chen-Yu Su
Dept. of CSIE, Chaoyang University of Technology, Taiwan, R.O.C.
shwu@cyut.edu.tw, s9427617@cyut.edu.tw

Tian-Jian Jiang, Wen-Lian Hsu
Institute of Information Science, Academia Sinica, Taiwan, R.O.C
Department of Computer Science, National Tsing-Hua University, Taiwan, R.O.C
tmjiang@iis.sinica.edu.tw, hsu@iis.sinica.edu.tw

Abstract

Many grammar checkers in rule-based approach do not handle errors that come from various usages, for example, the usages of prepositions. To study the behavior of prepositions, we introduce the language model into a grammar-checking task. A language model is trained from a large training corpus, which contains many short phrases. It can be used for detecting and correcting certain types of grammar errors, where local information is sufficient to make decision. We conduct several experiments on finding the correct English prepositions. The experiment results show that the accuracy of open test is 71% and the accuracy of closed test is 89%. The accuracy is 70% on TOEFL-level tests.

Keywords: *Language Model, Grammar Checker, English preposition usage*

1. Introduction

Computer-Aided Language Learning is a fascinating area; however, the computer still lacks many abilities of a human teacher, for example, the ability of grammar checking. Technically, it is hard to build a grammar checker that can deal with all types of errors. There are errors caused beyond the knowledge of syntax. For example, to overcome the misusing of prepositions, a system requires more semantic knowledge.

There are three major approaches to implement a grammar checker. The first strategy is the syntax-based checking [Jensen et al., 1993]. In this approach, a sentence is parsed into a tree structure. A sentence is correct if it can be parsed completely. Another choice is the statistics-based checking [Attwell, 1987]. In this approach, the system built a list of POS tag sequences based on a POS-annotated corpus. A sentence with known POS tag sequence is considered as a correct one. The last one is the rule-based checking [Naber 2003], where a set of rules is built manually and used to match against a text. Park et al. proposed an online English grammar checker for students who take English as the second language. This system focuses on a limited category of frequently occurring grammatical mistakes in essays written by students. The grammar knowledge is represented in Prolog language. [Park 1997]

We find that most grammar checkers do not deal with the errors of preposition usage. We suppose that it should be hard to write rules for all of the prepositions. To evaluate this difficulty, we introduce the language model into the grammar-checking task. Since a language model is usually trained from a large training corpus, it may contain many short phrases with prepositions.

The Language Model (LM) is one of the popular natural language processing technology for various applications, like information retrieval, handwriting recognition, speech recognition, and

machine translation. [Jurafsky and Martin, 2000] [Manning and Schutze, 1999] An LM uses short history to predict the next word. Word prediction is an essential subtask of speech recognition, handwriting character recognition, augmentative communication for the disabled, and spelling error detection. An LM can estimate the probability of a sentence. Therefore, it can be a way to distinguish good usages from bad ones of English prepositions.

Figure 1 shows a general architecture of an English grammar checker. An ideal system should consist of both rule-based and language model approaches. Linguistic knowledge of the rule-based system is acquired from domain experts. Statistical knowledge of the language model is gathered from training corpus by programs. In this paper, we design several experiments to assess the ability of the LM on the preposition usage problem.

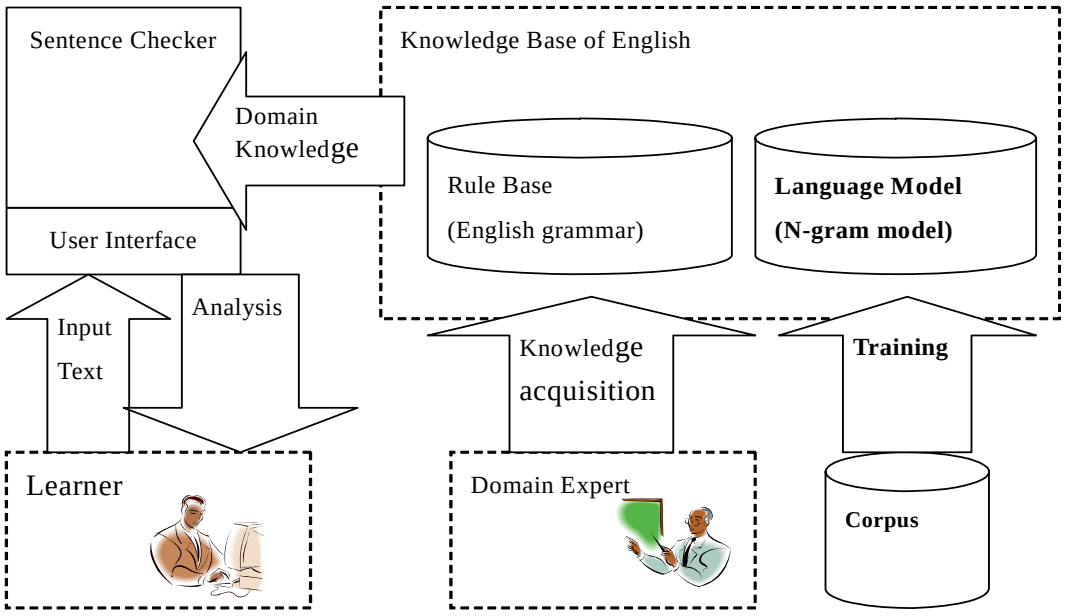

Figure1. The Architecture of a general English Grammar Checker

2. Statistical language model

We briefly restate the notation of N-gram language model. In this model, a sentence is viewed as a sequence of *n* words. The probability of a sentence in a language, say English, is defined as the probability of the sequence.

$$P(w_1^n) \equiv P(w_1, w_2, ..., w_n)$$

That can be further decomposed by the chain rule of conditional probability under the Markov assumption.

$$\begin{aligned} P(w_1^n) &= P(w_1)P(w_2 \mid w_1)P(w_3 \mid w_1^2)...P(w_n \mid w_1^{n-1}) \\ &= P(w_1)\prod_{k=2}^{n} P(w_k \mid w_1^{k-1}) \end{aligned}$$

Since it is not possible to collect all the history, a prefix of size N, as an approximation, is used to replace each component in the product.

$$P(w_n \mid w_1^{n-1}) \approx P(w_n \mid w_{n-N+1}^{n-1})$$

Usually, the N is 1, 2, or 3, are named as unigrams: $P(w_n)$, bi-grams: $P(w_n|w_{n-1})$, and tri-grams: $P(w_n| w_{n-1}w_{n-2})$ model, respectively.

Next step is to estimate the n-gram approximation from corpus. The basic way is called Maximum Likelihood Estimation (MLE), which calculates the relative frequency and is used as the estimation of probability. For bi-gram:

$$P(w_n \mid w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

And, for n-gram

$$P(w_n \mid w_{n-N+1}^{n-1}) = \frac{C(w_{n-N+1}^{n-1} w_n)}{C(w_{n-N+1}^{n-1})}$$

where *C* represents the count of each specified n-grams *w* in the corpus. MLE works well for high-frequency n-gram; however, no matter how large the corpus is, there are always some low-frequency n-grams. The frequency might be very low even zero. Some zeroes are really zeroes, which means that they represent meaningless word combinations. However, some zeroes are not really zeroes. They represent low frequency events that simply did not occur in the corpus and might exist in real world. When using n-gram model, we cannot assign a probability to a sequence where one of the component n-gram has a value of zero. An alternative solution is to smooth the probability estimations so that no component in the sequences is given a probability of zero.

2.1 Smoothing methods

To cope with the problem of unseen data, several smoothing methods are developed [Goodman, 2002]; they can be classified as discounting methods and model combination methods. Discounting methods adjust the probability estimators, so that zero relative frequency in the training data does not imply zero relative counts. Model combination methods combine available models (unigram, bi-gram, tri-gram, etc.) by interpolation and back-off. To our knowledge, Good-Turing discounting, absolute discounting and Chen-Goodman modified Kneser-Ney discounting are three of best smoothing methods; therefore, we use them in our experiments. [Chen and Goodman, 1998]

2.1.1 Good-Turing Discounting (GT)

Good-Turing discounting adjusts the count of n-gram from *r to* r^*, which is base on the assumption that their distribution is binomial [Good, 1953].

$$r^* = (r+1)\frac{N_{r+1}}{N_r} \qquad r < M$$

where N_r is types of n-gram occurring r times, and M is a threshold usually smaller than 5. Note that for r=0,

$$r^* = \frac{N_1}{N_0}$$

where N_0 is the number of n-grams that never occurred. The discounted probabilities are thus:

$$P_{GT}(w_1 \ldots w_n) = \frac{r^*}{N}$$

The Good-Turing formula only applies to the situation when r < 5, and need to renormalize to ensure that everything sums to one.

2.1.2 Absolute Discounting (AD)

In the absolute discounting model, all non-zero frequencies are discounted by a small constant discount rate *b*. And all the unseen events gain the frequency uniformly. [Ney et al., 1994]

$$N_0 \cdot P_0 = \frac{1}{N}\sum_{r=1}^{R} N_r \cdot discount\_rate = b \cdot \frac{K - N_0}{N},$$

Where *R* is the highest frequency and *K* is the number of bins that training instances are divided into:

$$K = \sum_{r=0}^{R} N_r, \qquad 0 < b \le 1$$

So the probability is

$$p_{abs}(w_1 \ldots w_n) = \begin{cases} \frac{r-b}{N}, 0 < r \le R \\ b \cdot \frac{K - N_0}{N \cdot N_0}, r = 0 \end{cases}$$

2.1.3 Modified Kneser-Ney discounting (mKN)

The Kneser-Ney discounting model is a back-off model based on an extension of absolute discounting which provides a more accurate estimation of the distribution. Chen and Goodman proposed a modified Kneser-Ney(mKN) discounting model. Instead of using a single discount for all nonzero counts as in KN smoothing, the mKN has three different parameters, D_1, D_2, and D_3 that are applied to n-grams with one, two, and three or more counts, respectively. The formula of mKN discounting is:

$$P_{mKN}(w_i \mid w_{i-n+1}^{i-1}) = \frac{c(w_{i-n+1}^{i}) - D(c(w_{i-n+1}^{i}))}{\sum_{w_i} c(w_{i-n+1}^{i})} + \gamma(w_{i-n+1}^{i-1}) P_{mKN}(w_i \mid w_{i-n+2}^{i-1})$$

where

$$D(c) = \begin{cases} 0 & if\ c = 0 \\ D_1 & if\ c = 1 \\ D_2 & if\ c = 2 \\ D_3 & if\ c \geq 3 \end{cases}$$

$$D_1 = 1 - 2\frac{N_1}{N_1 + 2N_2} \cdot \frac{N_2}{N_1}$$
$$D_2 = 1 - 3\frac{N_1}{N_1 + 2N_2} \cdot \frac{N_3}{N_2}$$
$$D_3 = 1 - 4\frac{N_1}{N_1 + 2N_2} \cdot \frac{N_4}{N_3}$$

and the gamma is a normalization constant such that the probabilities sum to one.

2.2 Entropy and Perplexity

Entropy is widely used to measure information. The entropy of a random variable X ranges over what are predictable set T (words, letters, or parts-of-speech) can be defined as:

$$H(X) = -\sum_{x \in T} p(x) \log_2 p(x)$$

Perplexity is a variant of entropy. Generally, the perplexity can be defined as:

$$2^{H}$$

Entropy of sequence of words can be defined as:

$$H(w_1, w_2, ..., w_n) = -\sum_{W_1^n \in L} p(W_1^n) \log_2 p(W_1^n)$$

Where $p(W_1^n)$ can be replaced by n-gram models.

3. Experiments

To assess the ability of how LM finds the right preposition, we use various sizes of training sets, and three test sets from three different sources.

3.1 Experiment design

For each original test sentence, we make up some wrong ones, and then calculate the perplexity of the test sentences. The perplexity is the measurement of how well the LM can predict the sentence. The sentence with the lowest perplexity is the most possible sentence with respect to the given LM; we assume that sentence is the correct one.

We conduct the experiments with the SRI Language Modeling Toolkit. [Stolcke, 2002] [http://www.speech.sri.com/projects/srilm/] The first test set comprises 100 sentences that we select from the training set. This test is regarded as a closed test. The second test set is another 100 sentences that we collect from various English literatures outside the training set. This is an open test. In the first two experiments, we focus on only three prepositions: in, on, and at. We fabricate the wrong sentences by replacing the correct preposition with other ones. The third test

set consists of 100 sentences of TOFEL-level questions. We collect these sentences from TOFEL reference books; they contain most of the English prepositions.

The training corpus is selected from LDC Gigaword corpora [LDC 2003]. The Gigaword corpora are very large English newswire text collections. There are four distinct international sources: Agence France Press English Service (AFE), Associated Press Worldstream English Service(APW), The New York Times Newswire Service (NYT) and The Xinhua News Agency English Service (XIE). The total size of the corpora is more than one gigabyte in word counts.

We use the NYT corpus as the training set. The training set sizes in different experiments are different. For bi-gram model, we select the news of the NYT from January 1999 to June 2002 as our training corpus. It consists of 351,427,489 words and is about 1.89 GB. We do not perform any preprocessing and do not remove stop words. For tri-gram model, we select the news of NYT from January 2001 to June 2002. This corpus consists of 156,896,511 words and the size is about 856 MB.

Table 1 The sizes of the training sets for Bi-gram model

Training Set	# of words	MB
nyt200111-200206(8)	69865209	384
nyt200101-200206(18)	156896511	865
nyt199901-200206(42)	351427489	1890

Table 2 The sizes of the training sets for Tri-gram model

Training Set	# of words	MB
nyt200203(1)	9310195	52
nyt200203-200204(2)	18734690	102
nyt200201-200206(6)	52574963	289
nyt200108-200206(11)	97578257	537
nyt200101-200206(18)	156896511	865

3.2 Experiment results

3.2.1 Closed tests

In the first experiment, we select 100 sentences from our training corpus as the test set. We fabricate the wrong sentences by replacing the correct preposition with other prepositions. We calculate the perplexity of the sentences with LMs and check if the sentence with the lowest perplexity is the original one. We do not list the values of perplexity, since it is meaningless for the closed test. In computing perplexities, the model must be constructed without any knowledge of the test set. The knowledge of the test set will make the perplexity artificially low.

Table 3 and 4 shows the accuracy of the first test set on various LMs. In this task, the test accuracy of bi-gram is lower than that of tri-gram; even the training size is doubled. The accuracy for tri-gram converges as the size of training set increasing. In Table 4, we enlarge the training corpus from 1-month news articles to 18-months news articles for the training set, the test accuracy does not increase much. The mKN smoothing method gives the best accuracy 89%.

Table 3. The closed test accuracy of bi-gram models on various training sets.

	Smoothing method		
Training Set	GT	mKN	AD
nyt200111-200206(8)	65%	65%	**73%**
nyt200101-200206(18)	65%	65%	
nyt199901-200206(42)	66%		

Table 4. The closed test accuracy of tri-gram models on various training sets.

	Smoothing method		
Training Set	GT	mKN	AD
nyt200203(1)	80%	86%	88%
nyt200203-200204(2)	80%	85%	
nyt200201-200206(6)	87%	88%	
nyt200108-200206(11)	85%	88%	
nyt200101-200206(18)	85%	**89%**	

3.2.2 Open tests

In the second experiment, the test set is another 100 sentences that we collect from the following five English literary works: (download from Project Gutenberg Online Book Catalog http://www.gutenberg.org/)

1. Amusements in Mathematics by Henry Ernest Dudeney.
2. Grimm's Fairy Tales by Jacob Grimm and Wilhelm Grimm.
3. The Art of War by Sun-Zi.
4. The Best American Humorous Short Stories.
5. The War of the Worlds by H. G. Wells.

Again, we fabricate the wrong sentences by replacing the correct preposition with other prepositions. We calculate perplexities of the sentences with LMs of different sizes and check if the sentence with the lowest perplexity is the original one.

Table 5 and 6 show the accuracy of the second test set. The mKN smoothing method gives the best accuracy 71%.

Table 5. The open test accuracy of bi-gram models on various training sets.

	Smoothing method		
Training Set	GT	mKN	AD
nyt200111-200206(8)	47%	49%	50%
nyt200101-200206(18)	48%	**51%**	
nyt199901-200206(42)	47%		

Table 6. The open test accuracy of tri-gram models on various training sets.

	Smoothing method		
Training Set	GT	mKN	AD
nyt200203(1)	61%	57%	61%
nyt200203-200204(2)	61%	62%	
nyt200201-200206(6)	67%	69%	
nyt200108-200206(11)	68%	69%	
nyt200101-200206(18)	68%	**71%**	

3.2.3 TOEFL-level tests

There is a problem in the setting of the previous two experiments. We do not check if the fabricated wrong sentences are also legal in the real world. Therefore, we collect 100 TOEFL-level single-choice questions from pseudo TOEFL tests. Each sentence has a blank for a preposition. Four candidates are available, but only one is correct. For example:

My sister whispered __ my ear.

(a) in (b) to (c) with (d) on

Then our task is to distinguish which of the following four sentences is correct.

My sister whispered in my ear. (correct)

My sister whispered to my ear. (wrong)

My sister whispered with my ear. (wrong)

My sister whispered on my ear. (wrong)

We also train our LMs with different sizes of training set. We then use the LMs to calculate the perplexities of the four sentences. The system regards the sentence with the lowest perplexity as the correct one. The results in Table 7 show that tri-gram model with mKN smoothing gives the best result even though the training size is much smaller than the one for the bi-gram model.

Table 7. The TOEFL-level tests accuracy of bi-gram and tri-gram model

	Smoothing method	
Training Set	GT	mKN
Bigram model nyt199901-200206(42)	53%	54%
Trigram model nyt200101-200206(18)	69%	**70%**

3.3 Error Analysis

Table 8 shows a part of the test results that the LM gives wrong answers. The system chooses the candidate with the lowest perplexity as the answer; however, in these cases, the candidates with the lowest perplexities are wrong. We manually check these sentences and identify the necessary keyword. We find that, to give the right answer, the system must refer to some words that are not close to the blank. Such long-distance features cannot be learned in a short windows size of two or three; therefore, the tri-gram model cannot give the right answer.

Table 8. Error examples of using the tri-gram model on TOEFL-level tests, where the logprob is the logarithm of n-gram probability and the perplexity is defined as 10^(-logporb/ # of words in the sentence).

No.	Question	choices	correct answer	logprob	perplexity	LM answer
1	It is sometimes difficult to make pleasant conversation ___ people you have just met.	among		-35.5006	343.367	
		to		-33.5936	250.923	v
		for		-36.7712	423.168	
		with	v	-33.6707	254.127	

2	I have no knowledge whatever ___ the sciences.	of	v	-23.0051	751.007	
		to		-23.5853	887.482	
		in		-20.978	419.037	v
		on		-23.8697	963.202	
3	I'm bored ___ staying here.	of	v	-16.5306	2023.56	
		in		-16.7524	2241.21	
		with		-15.1587	1075.83	v
		for		-16.5074	2002.09	
4	He lives ___ 144 Wall Street.	at	v	-17.7392	904.767	
		in		-15.9947	463.223	v
		on		-18.7051	1310.76	
		by		-18.3593	1147.85	
5	We danced ___ the music of Jimmy Dorsey's band.	to	v	-25.8146	738.4	
		with		-26.2586	827.221	
		in		-25.5648	692.674	v
		on		-26.0885	791.996	
6	Write your composition ___ ink.	in	v	-19.8675	9408.04	
		on		-21.0123	15938.9	
		with		-19.4143	7635.85	v
		by		-20.1884	10906.4	
7	In a short while, I'll be free ___ all my worries.	with		-30.8353	635.642	
		of	v	-28.7124	407.583	
		about		-32.6347	926.383	
		to		-27.3856	308.745	v
8	He stopped the car ___ the park.	by	v	-16.5065	228.069	
		in		-13.9978	99.9273	v
		on		-15.4607	161.688	
		to		-14.4585	116.279	
9	That would be ___ my dignity.	beneath	v	-15.0318	320.109	
		under		-14.2876	240.581	
		beyond		-13.86	204.171	v
		above		-15.9486	455.096	
10	The fire began ___ the fifth floor of the hotel, but it soon spread to adjacent floors.	on	v	-40.8443	252.701	
		in		-39.2952	204.872	v
		at		-41.4813	275.47	
		of		-44.1101	393.292	
11	The main office of the factory can	in		-32.6532	109.856	

	be found ___ Maple Street in New York City.	at		-32.1039	101.507	v
		on	v	-33.8979	131.408	
		from		-34.4493	142.26	
12	Conifers first appeared on the Earth ___ the early Permian period, some 270 million years ago.	when		-42.6762	699.972	
		or		-41.3285	569.158	
		and		-39.3227	418.322	v
		during	v	-40.0914	470.714	
13	She'll be here ___ about twenty minutes.	by		-22.3606	1564.51	
		on		-21.2314	1079.08	v
		at		-21.4576	1162.45	
		in	v	-22.4876	1631.24	

4. Conclusions and discussions

In this paper, we report the evaluation of adopting the language model on checking the English prepositions. In our experiments, we assume that a correct sentence has less perplexity than the wrong ones. The experiment results show that tri-gram language model can find most of the correct prepositions. The modified Kneser-Ney smoothing method gives the best accuracy in three test sets. Experiment results show that the accuracy of open test is 71%, the accuracy of closed test is 89%, and the accuracy on TOEFL-level test is 70%. This approach has two advantages, the first one is that it requires only untagged corpus. The second one is that it requires no domain knowledge. Thus, the approach can cooperate with other approaches in the future easily.

To improve the accuracy, the system requires more linguistic knowledge. Other feature-based machine learning approaches, for instance, Maximum Entropy (ME) [Berger et al., 1996], Conditional Random Fields (CRF) [Lafferty et al., 2001] are also promising. They can incorporate more long-distance linguistic features that LM cannot. [Rosenfeld, 1997].

The collection of linguistic features requires more knowledge engineering. In an English grammar textbook of college-level [Eastwood, 1999], the usages of the prepositions are addressed by rules and examples, as listed in Table 9 and 10. To cooperate with the rules, a system requires linguistic resources to recognize the names of different entities such as countries, regions, towns, and time expressions. Moreover, the system still requires templates of specific usages. Table 10 gives many common phrases examples of the three prepositions: in-on-at (used for place only). These "common" phrases might appear in the corpus many times. Since they are short, they will be in the tri-gram model.

Table 9. Rules of preposition usage [Eastwood, 1999]

	Positive and Negative Rules
At	1. Use in (not at) before the names of countries, regions, cities, and large towns. 2. Use in (not at) with seasons, months, and years. 3. Use on (not at) before dates. 4. Without at before 'an hour before', 'a week later', 'two years afterwards' 5. Do not use at to introduce a time expression with ago.
In	1. on a day or date, not in 2. in the morning/afternoon/evening' but 'the following morning', 'the next afternoon', 'the previous evening', etc. 3. When talking about how long something lasts or continues, use for, not in. 4. on/upon doing something, not in 5. made of wool/wood etc., not in 6. in is not used in expressions such as 'the shop is open six days a week.' 'He visits his father three times a year.' 'Bananas cost fifty pence a pound.' 'I drove to the hospital at ninety miles an hour.'
On	1. Do not use a preposition to begin a time expression with next when the point of time is being considered in relation to the present: 'the next morning', 'the next afternoon'. 2. a good/bad thing about someone/something, not on 3. When talking about a particular afternoon, use on. When speaking generally, use in.

Table 10. Common phrase for in, on, and at [Eastwood, 1999]

	Common phrases (place)
In	In prison/hospital In the lesson In a book/newspaper

	In the photo/picture In the country In the middle In the back/front of a car In a queue/line/row
On	On the platform On the farm On the page/map On the screen On the island/beach/coast Drive on the right/left On the back of an envelope
At	At the station/airport At home/work/school At the seaside At the top/bottom of a hill At the back of the room At the end of a corridor

Acknowledgement

This research was partly supported by the National Science Council under GRANT NSC 94-2218-E-324 -003.

References

[Atwell 1987] Eric Atwell, Stephen Elliott, Dealing with ill-formed English text, in: The computational analysis of English : a corpus-based approach / edited by Roger Garside, Geoffrey Leech, Geoffrey Sampson, The computational analysis of English, London ; New York, Longman, 1987.

[Berger et al., 1996] A. Berger, S. A. Della Pietra, and V. J. Della Pietra, A maximum entropy approach to natural language processing, Computational Linguistics, vol. 22, pp. 39-71, 1996.

[Chen and Goodman, 1998] S. F. Chen and J. Goodman, An empirical study of smoothing techniques for language modeling, Technical Report TR-10-98, Computer Science Group, Harvard University, Aug. 1998.

[Eastwood, 1999] John Eastwood, Oxford Practice Grammar, Oxford University Press, 1999.

[Good, 1953] I.J. Good, The population frequencies of species and the estimation of population parameters, Biometrika 40: pp 237-264, 1953.

[Goodman, 2002] Joshua T. Goodman, A bit of Progress in Language Modeling, Technical Report, MSR-TR-2001-72, Microsoft Research, Redmond, 2002.

[Jensen et al.,1993] Karen Jensen, George E. Heidorn, Stpehen D. Richardson (Eds.): Natural language processing: the PLNLP approach, Kluwer Academic Publishers, 1993.

[Jurafsky and Martin, 2000] Jurafsky, D. and Martin, J.H., Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition, Prentice-Hall, Upper Saddle River, New Jersey, 2000.

[Lafferty et al, 2001] Lafferty, J., McCallum, A., and Pereira, F., Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. Paper presented at the ICML-01.

[LDC 2003] LDC, Gigaword Corpora. http://wave.ldc.upenn.edu/Catalog/CatalogEntry.jsp?catalogId=LDC2003T05

[Manning and Schutze, 1999] Chris Manning and Hinrich Schütze, Foundations of Statistical Natural Language Processing, MIT Press, Cambridge, MA: May 1999.

[Naber et al., 2003] Daniel Naber, A Rule-Based Style and Grammar Checker, diploma thesis, University Bielefeld, 2003.

[Ney et al., 1994] Hermann Ney, Ute Essen, and Reinhard Kneser, On structuring probabilistic dependencies in stochastic language modeling, Computer Speech and Language 8: pp1-28, 1994.

[Park et al., 1997] Jong C. Park, Martha Palmer, and Clay Washburn, An English Grammar Checker as a Writing Aid for Students o f English as a Second Language, in Proceedings of the Fifth Conference on Applied Natural Language Processing, 1997. http://acl.ldc.upenn.edu/A/A97/A97-2014.pdf

[Rosenfeld, 1997] Ronald Rosenfeld, A Whole Sentence Maximum Entropy Language Model, In Proc. IEEE workshop on Automatic Speech Recognition and Understanding, Santa Barbara, California, December 1997.

[Stolcke, 2002] Andreas Stolcke, SRILM - An Extensible Language Modeling Toolkit, in Proc. Intl. Conf. Spoken Language Processing, Denver, Colorado, September 2002