Working Paper Series
ISSN 1170-487X

Tag Based Models of English Text

by W J Teahan and John G Cleary

Working Paper 97/24
November 1997

Department of Computer Science
The University of Waikato
Private Bag 3105
Hamilton, New Zealand

Tag based models of English text

W. J. Teahan, John G. Cleary[1]

Department of Computer Science, University of Waikato, New Zealand

The problem of compressing English text is important both because of the ubiquity of English as a target for compression and because of the light that compression can shed on the structure of English. English text is examined in conjunction with additional information about the parts of speech of each word in the text (these are referred to as "tags"). It is shown that the tags plus the text can be compressed more than the text alone. Essentially the tags can be compressed for nothing or even a small net saving in size. A comparison is made of a number of different ways of integrating compression of tags and text using an escape mechanism similar to PPM. These are also compared with standard word based and character based compression programs. The result is that the tag character and word based schemes always outperform the character based schemes. Overall, the tag based schemes outperform the word based schemes. We conclude by conjecturing that tags chosen for compression rather than linguistic purposes would perform even better.

1 Tag based compression

The basis of modern high performance compression is the adaptive use of prior contexts to predict the next item. For example, when compressing English text a word or character is predicted on the basis of the immediately preceeding word or character. The predictions are built up adaptively as more text is seen so that latter predictions depend on all the text that has preceeded them. The compressors with the best general performance reported in the literature are all of this adaption plus prediction form (Cleary & Teahan, 1997; Bunton, 1996; Burrows & Wheeler, 1994) and they have been used to successfully compress data as diverse as graphics files, geophysical data records and computer executables.

In this paper we consider a more specific problem, that of compressing English text. This differs from the general compression problem because much is known *a priori* about the structure of English. It should be possible to use this structure to achieve better compression. One way that this has been done is to use the fact that English can be segmented into words and to use words rather than characters as the fundamental unit of compression. This is found to be faster (because less encoding operations are necessary) and to achieve up to 4% better compression than purely character based models (Teahan, 1997).

Another approach to modelling adopted by Teahan & Cleary (1997) is to use parts-of-speech (tags) such as *noun*, *verb*, and *adjective*. Their approach is explored in more detail in this paper. The idea is that knowing the tag of a word helps in predicting it. The advantage of using the tag is that it may have occurred many times previously. Hence, a good representative sample of what is likely to follow it

[1]email {wjt, jcleary}@waikato.ac.nz

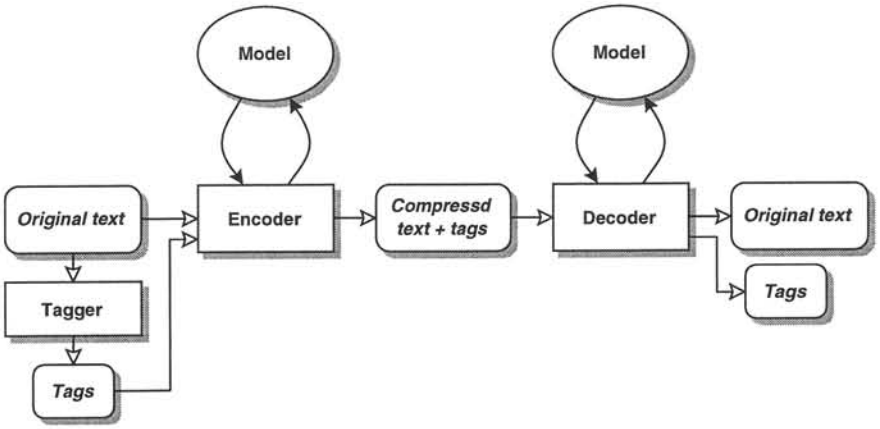

Figure 1: Using a tagger to compress English text.

has been built up. By contrast, an individual word may have occurred only a small number of times. Traditional language modelling approaches (for example, used in speech recognition and machine translation) have been either word or part of speech based (Brown *et al.*, 1992b; Jelinek, 1990; Kuhn & De Mori, 1990). Results with these models have shown that the word based approach generally performs better.

There are two major issues with using tags: first, the words in the text must have tags assigned to them somehow; and second, the tags need to be encoded in the models along with the text itself. This has the potential for increasing the size of the compressed text. However, the extra contextual information provided by the tags more than compensates for this and we will see that the total result is slightly better than pure word based coding.

For the models we are concerned with, we assume that the text has already been tagged using a much more comprehensive tag set (such as those shown in Tables 2 and 3) and we wish to explicitly encode and decode these tags along with the words (as shown in Figure 1).

The next section describes two adaptive models, one word based and the other tag based, that have been found to perform better than other models in practice. Following that, results of experiments with compressing English text are discussed. These are split into two subsections—results with manually tagged texts, and results with texts automatically tagged by computer.

2 Tag and word based models

Both the tag and word based models recommended here exploit the blending mechanism of the PPM compression scheme (Cleary & Witten, 1984; Cleary, Teahan & Witten, 1995; Cleary & Teahan, 1998). Higher order contexts are tried first, but if the next word has not been seen before in this context then a lower order context is used instead. So that the decoder knows which context to use, an "escape" symbol is transmitted to signal that the prediction should be done with a lower order context.

Experiments reported in Teahan & Cleary (1996) show that a simple escape strategy performs best in most cases. This method estimates the probability of the

WW model	WTW model	TTWT model
$p(w_i \mid w_{i-1})$	$p(w_i \mid t_i w_{i-1})$	$p(t_i \mid t_{i-1} w_{i-1} t_{i-2})$
$\hookrightarrow p(w_i \mid)$	$\hookrightarrow p(w_i \mid t_i)$	$\hookrightarrow p(t_i \mid t_{i-1} w_{i-1})$
$\hookrightarrow$ *character model*	$\hookrightarrow$ *character model*	$\hookrightarrow p(t_i \mid t_{i-1})$
		$\hookrightarrow p(t_i \mid)$
		$\hookrightarrow p_{eq}(t_i \mid)$

Figure 2: Some models for predicting tags and words

escape symbol as being proportional to the number of words in the context which have occurred only once *i.e.* the number of singletons. Performance of these models can be improved further by two simple mechanisms—the first, called *update exclusions*, updates the counts only in contexts that actually make the prediction. The second, called *full exclusions*, excludes words already predicted by higher order contexts. Both update and full exclusions typically improve the compression by a few per cent.

The representation of the two best performed models experimented with are shown in Figure 2. The order 1 word model (labelled "WW") first predicts the word using just the previous word, but escapes to an order 0 model if the word is not predicted, then to a character based model (a fixed order PPM model) if the word is not predicted at all. In the diagram, the symbol $\hookrightarrow$ represents the escape process.

The second model shown in Figure 2 (labelled "WTW") first predicts the word using the current tag and the previous word. If this is unsuccessful, it tries based on the current tag only, otherwise it escapes down to the character based model. This model must include some mechanism for predicting the tags as well as the words. The best model we have found for this based on results from compression experiments is labelled "TTWT"—it first uses the prior tag, the prior word and the tag preceding that to predict the tag. If unsuccessful, it uses the escape hierarchy shown.

These models are described in more detail in Teahan & Cleary (1996). An efficient trie-based data structure that maintains the cumulative frequency counts required for arithmetic coding for these models is also described there.

3 Compression experiments

Compression experiments for these models were conducted on various texts (both tagged and non-tagged). All experiments were with texts converted to 27 character English—26 letters plus space. For these experiments, a "word" was considered to be any consecutive sequence of letters between spaces. Tags were assigned to words using the corresponding tag in the tagged text—if a word was split into more than one part (if the original word was hypenated, for example), then the same tag was assigned to each new part *e.g.* *Vice-Chairman*/**NN** becomes *vice*/**NN** *chairman*/**NN**.

The compression experiments are split into two sections—experiments with tagged corpora where the tags have been manually checked; and experiments with texts where the tags have been assigned automatically by computer. Compression ratios are shown in bits per character (bpc). More details of these and other experiments may be found in Teahan (1997).

Tagged text	Number of words	PPMD5 + bigram (bpc)	WW (bpc)	WTW + TTWT (bpc)
LOB Corpus:				
• *all* 5636660 *characters*	1021049	1.860	1.783	**1.781**
• *last* 10001 *chars. using the preceding text for training*	1912	1.863	1.809	**1.784**
Wall Street Journal:				
• *all* 15398849 *characters*	2614956	1.602	**1.539**	1.547
• *last* 10010 *chars. using the preceding text for training*	1736	1.553	**1.490**	**1.490**

Table 1: How well the models compress the manually tagged texts

3.1 Experiments with manually tagged texts

Compression results on manually tagged corpora are summarized in Table 1. Two pre-tagged text corpora were obtained for the experiments—the LOB Corpus (Johansson *et al.*, 1986) and the Wall Street Journal (ACL-DCI, 1991). Table 1 compares how well the tag and word based models perform at compressing these texts with the best of the character based models, labelled "PPMD5+bigram." It combines an order 5 PPM model with bigram coding as described in Teahan & Cleary (1996). Bigram coding replaces frequently occurring character bigrams—two letter sequences—with a single unique code. They showed that it typically improves PPM compression on English text by up to 7%. Also listed are results for a sample taken from the last 10,000 characters or so of each text[2] using the preceding text for training.

The results show that performance of the tag based methods is comparable with the best word based models. This is surprising as the tag-based model has to encode *both* the tags and the words—the tags are produced by the decoding process at no extra cost. No attempt was made to optimize the performance of the tag based models. The tags used were designed primarily for linguistic purposes, and further gains should be possible by optimizing the tag set to improve compression performance. Both the word and tag based models are better than the character based method by 3 to 4%.

Figures 3 and 4 show how training text improves compression for the WTW+ TTWT model for the sample text at the end of the LOB Corpus and the Wall Street Journal. Also included are the curves (shown by the dashed lines) for the best of the character and word based models. They show that the initial performance of the tag based model is poor in comparison to the other two models but its rate of improvement is better, gaining parity with the word model at about the 1×10^5 character mark. It is unclear whether this trend would continue if more training text was available.

[2]The size of the sample was increased slightly (to 10,001 characters for the LOB Corpus, and 10,010 characters for the Wall Street Journal) to ensure that the first character in the sample was at the beginning of a word.

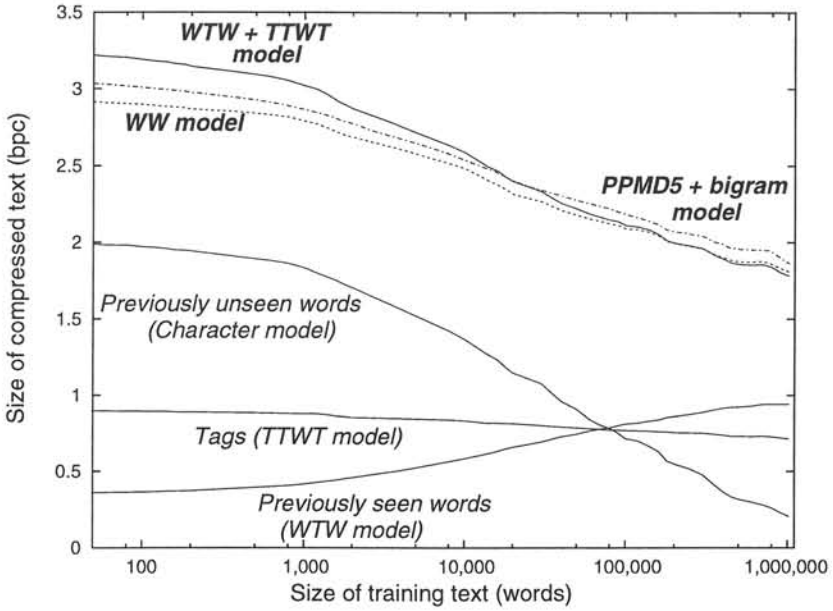

Figure 3: How training improves compression for the last 10001 characters of the LOB Corpus

Three further curves are plotted that track the costs of encoding the tags (TTWT model), the previously seen words (WTW model) and the previously unseen words (character model). When added together, these three costs equal the overall cost for the WTW+TTWT model. The main contributing factor to the improvement in compression is the reduction in the number of unseen words with larger training texts. Consistent but slow improvement throughout is apparent for the TTWT model. The curve for the WTW model on the other hand steadily increases before plateauing out beyond 2×10^5 characters, and marginally decreasing in the case of the Wall Street Journal text. Experiments with other tag/word models show that the WW and WTW models are consistently the best two models for predicting the words.[3] Experiments also show that performance degrades with higher order models. It is unclear how much larger training texts and the addition of the blending mechanisms described in Bunton (1996) will affect these results.

3.2 Experiments with computer tagged texts

Experiments were also conducted on a number of computer-tagged texts—the Brown Corpus (Francis & Kučera, 1982), the LOB Corpus, the Wall Street Journal, the King James Bible, the complete works of Shakespeare and Jane Austin (these last three texts are available in the public domain), and the Jefferson Corpus.[4]

An important application of the tag based models is their ability to compare the

[3]For a comparison of the performance of several other models, see Teahan (1997).

[4]This corpus consists of text obtained by scanning into the computer the six volumes (3,069 pages) of Dumas Malone's *Jefferson and his time* (1977). Teahan & Cleary (1996) first used this text to arrive at a machine estimate of 1.48 bpc for the "entropy" of English based on PPM character based models.

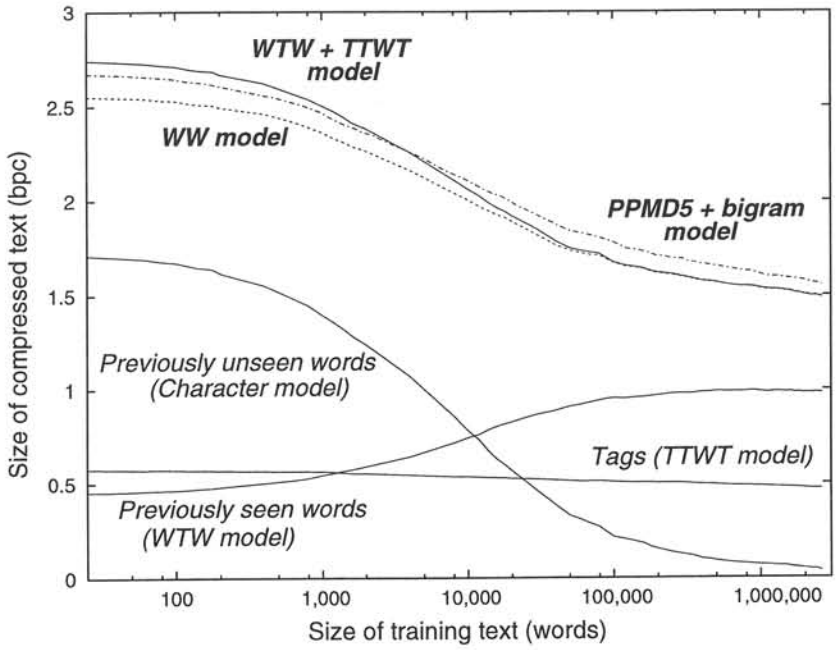

Figure 4: How training improves compression for the last 10010 characters of the Wall Street Journal

performance at word prediction of different taggers and tag sets (whether the tags are assigned manually or by computer). An accuracy level of 95–97% is typically reported for computer taggers (Brill, 1994; Charniak, 1993; Church & Mercer, 1993).[5] These taggers perform well at automatically tagging text—most of the tagging errors are caused by words that are unseen or rarely seen in the training corpora. Consequently, computer tagged texts should work well with these models, and we will see that this is true in practice.

An important issue with using computer tagged texts is the tag set chosen to tag the text. The AMALGAM[6] project (Atwell, Hughes & Souter, 1994) has developed a tagging program accessed over the Internet to tag text with up to 8 annotation schemes. Table 2 provides a description, the number of tags for each and an identifying label (used for reference in the following discussion). These represent the main tag sets that have been adopted in various research corpora.

In an experiment, the AMALGAM tagger was used to tag the Jefferson text with each of the eight tag sets. The tags assigned by the tagger to the opening line of Dumas Malone's *Jefferson the Virginian* are shown Table 3. Four of the tag sets are similar—BROWN, LOB, PENN and SEC—as the latter three were based on the first. The SEC tag set is essentially the LOB tag set with minor changes. The other

[5]Samuelson & Voutilainen (1997) highlight problems with using accuracy levels to compare taggers. They claim higher levels of accuracy are possible, but Church disputes this, arguing that the 97% level is an "upper bound" because linguists performing the task manually disagree in at least 3% of all cases.

[6]The acronym stands for "Automatic Mapping Among Lexico-Grammatical Annotation Models." The tagger is based on Brill's (1993) rule-based tagger.

Label	Tags	Description/reference
BROWN	226	Brown Corpus tag set
ICE	205	International Corpus of English tag set
LLC	210	London-Lund Corpus tag set
LOB	153	Lancaster-Oslo/Bergen Corpus tag set
PARTS	19	Tag set used by the UNIX `parts` program
PENN	45	University of Pennsylvania Corpus tag set
POW	66	Polytechnic of Wales Corpus tag set
SEC	150	Spoken English Corpus tag set

Table 2: Tag sets provided by the AMALGAM tagger

tag sets are noticeably different from these four. The PARTS tag set, for example, is based on the tag set devised for the UNIX `parts` program.

Each of the tagged texts was compressed using the WTW+TTWT model with results shown in Table 4. Compression ratios (sorted in ascending order) are listed for two configurations: first, for all the text found in the six volumes of Dumas Malone's work (6448790 characters and 1113235 words) without any training text; and second, for the last chapter of *Jefferson the Virginian* (46142 characters and 7984 words) trained on the remaining text from Dumas Malone's work. These results are compared at the bottom of the table with the best character and word models (PPMD5+bigram and WW) and the standard PPMD5 model on the untagged text.

The results show that a number of the tag sets (BROWN, LOB, PENN and SEC) outperform the word based model, and all of them are better than the character based model. The performance of this tag set is surprising—it was designed before the other three (which supposedly have all been "improved" linguistically). Also interesting is the performance of the SEC tag set compared to the LOB tag set. The SEC tag set was trained solely on transciptions of spoken English. Atwell *et al.* (1994) point out that consequently "phenomena that are used primarily for English in its written form will not be found in SEC" (one example being abbreviations). The difference in quality of training data may also account for the differences observed between the other models.

Also shown in the table are the actual number of distinct tags found in the tagged texts (these are all notably less than the possible number of tags given in Table 2), and the compression ratios for encoding the three separate components of the overall model: the previously seen words (WTW model), the tags (TTWT model) and the previously unseen words (character model).

To further test the robustness of these results, other texts were tagged using the three best performing tag sets (BROWN, PENN and LOB) and then compressed. Table 5 shows that the performance of the tag models with the BROWN and PENN tag sets are comparable with the best word (WW) model over a diverse range of texts. The results for the LOB tag set are slightly worse, but all three tag sets are better than the best character model (PPMD5+bigram). The difference, however, between the best and worst models averaged over all the texts is still less than 4%.

An interesting comparison can be made between the results for the computer

Tag Set	AMALGAM-tagged text
Untagged	thomas jefferson was born in a simple wooden house in what is now albemarle county virginia
BROWN	thomas/**NP** jefferson/**NP** was/**BEDZ** born/**VBN** in/**IN** a/**AT** simple/**JJ** wooden/**JJ** house/**NN** in/**IN** what/**WDT** is/**BEZ** now/**RB** albermarle/**NP** county/**NN** virginia/**NP**
ICE	thomas/**N**(prop, sing) jefferson/**N**(prop, sing) was/**V**(cop,past) born/**ADJ**(edp) in/**PREP**(ge) a/**ART**(indef) simple/**ADJ** wooden/**ADJ** house/**N**(com,sing) in/**PREP**(ge) what/**PRON**(nom) is/**V**(cop,pres) now/**ADV**(ge) albermarle/**N**(prop,sing) county/**N**(prop,sing) virginia/**N**(com,sing)
LLC	thomas/**NP** jefferson/**NP** was/**VB+5** born/**VA+N** in/**PA** a/**TF** simple/**JA** wooden/**JA** house/**NC** in/**PA** what/**GC** is/**VB+3** now/**AC** albermarle/**NP** county/**NP** virginia/**NP**
LOB	thomas/**NP** jefferson/**NP** was/**BEDZ** born/**VBN** in/**IN** a/**AT** simple/**JJ** wooden/**JJ** house/**NN** in/**IN** what/**WDT** is/**BEZ** now/**RN** albermarle/**NP** county/**NPL** virginia/**NP**
PARTS	thomas/**adj** jefferson/**noun** was/**be** born/**adj** in/**prep** a/**art** simple/**adj** wooden/**adj** house/**noun** in/**prep** what/**pron** is/**be** now/**adv** albermarle/**adj** county/**noun** virginia/**noun**
PENN	thomas/**NNP** jefferson/**NNP** was/**VBD** born/**VBN** in/**IN** a/**DT** simple/**JJ** wooden/**JJ** house/**NN** in/**IN** what/**WP** is/**VBZ** now/**RB** albermarle/**NNP** county/**NNP** virginia/**NNP**
POW	thomas/**HN** jefferson/**HN** was/**OM** born/**AX** in/**P** a/**DQ** simple/**H** wooden/**AX** house/**H** in/**AX** what/**HWH** is/**OM** now/**AX** albermarle/**HN** county/**HN** virginia/**HN**
SEC	thomas/**NP** jefferson/**NP** was/**BEDZ** born/**VBN** in/**IN** a/**AT** simple/**JJ** wooden/**JJ** house/**NN** in/**IN** what/**WDT** is/**BEZ** now/**RN** albermarle/**NP** county/**NP** virginia/**NP**

Table 3: Tagging the opening line to Dumas Malone's *Jefferson the Virginian*

and manually tagged texts. The manually tagged LOB corpus requires 1.781 bpc to compress it; in comparison, the results for the computer tagged text (using the same LOB tag set) is only slightly worse (1.784 bpc). Even more interesting is the comparison for the Wall Street Journal—the computer tagged text (with the PENN tag set) compresses better than the manually tagged text (1.536 bpc compared to 1.547 bpc). These comparisons, however, are slightly biased in that part of the training corpora used to train the AMALGAM tagger includes parts of the texts being compressed (for example, the training corpus used to train the tagger for the LOB tag set includes 20% of the LOB corpus itself). Even so, the computer tagged models still do remarkably well compared with the manually tagged models.

Tag Set	Number of tags	WTW model (bpc)	TTWT model (bpc)	Char. model (bpc)	*All the text* (bpc)	*Last chapter* (bpc)
BROWN	102	0.775	0.549	0.109	**1.433**	**1.412**
PENN	34	0.864	0.465	0.112	1.441	1.426
LOB	119	0.745	0.586	0.114	1.445	1.427
SEC	101	0.741	0.592	0.121	1.454	1.432
LLC	123	0.730	0.614	0.121	1.465	1.442
ICE	141	0.733	0.611	0.126	1.470	1.446
POW	55	0.887	0.461	0.122	1.470	1.448
PARTS	12	0.938	0.409	0.128	1.475	1.458
PPMD5					1.620	1.598
PPMD5+bigram model					1.513	1.487
WW model					1.455	1.442

Table 4: Comparing the WTW+TTWT model performance for different tag sets on the AMALGAM-tagged text of the Jefferson Corpus

AMALGAM-tagged text	**PPMD5 +bigram** (bpc)	**WW model** (bpc)	**WTW+TTWT model** *BROWN* (bpc)	*PENN* (bpc)	*LOB* (bpc)
Jefferson Corpus	1.513	1.455	**1.433**	1.441	1.445
Brown Corpus	1.874	**1.797**	1.805	1.809	1.829
LOB Corpus	1.860	1.784	**1.781**	1.784	1.784
Wall Street Journal	1.603	1.542	1.539	**1.536**	1.554
King James Bible	1.464	1.439	1.423	**1.422**	1.432
Complete works of Shakespeare	1.931	**1.846**	1.892	1.888	1.933
Complete works of Jane Austen	1.607	1.534	**1.519**	1.523	1.531
Average	1.693	1.628	**1.627**	1.629	1.644
Weighted average	1.676	1.612	**1.611**	**1.611**	1.627

Table 5: Comparing model performance on various AMALGAM-tagged texts

4 Conclusions

A number of models have been investigated for compressing English text. The results show that models based on parts-of-speech (tags) can perform as well as word based models. These models require that *both* the tags and the words be encoded. Surprisingly, the tags are produced as a by-product of the decoding process at no extra cost.

An important application of these models is their ability to compare the performance of different taggers and tag sets at word prediction. Results show that tags assigned automatically by a computer tagger work as well as those manually checked by humans. The Brown Corpus tag set achieves the best overall performance on a diverse range of texts when compared to other commonly used tag sets.

No attempt was made to optimize the performance of these models. As the tags

used were designed primarily for linguistic purposes, further gains should be possible by optimizing the tag set to improve compression performance. Jelinek (1990) describes several tag and word based language models designed for speech recognition. He reports significant improvement in model performance using automatically derived parts of speech.

REFERENCES

ACL-DCI. 1991. Assoc. for Comp. Linguistics Data Collection Initiative CD-ROM 1.

Brill, E. 1994. "Some advances in rule-based part of speech tagging." *Proceedings of the Twelth National Conference on Artificial Intelligence (AAAI-94)*, Seattle, Washington.

Brown, P.F., Della Pietra, V.J., deSouza, P.V., Lai, J.C. and Mercer, R.L. 1992. "Class-based *n*-gram models of natural language." *Comp. Linguistics*, **18**(4), 467–479.

Bunton, S. 1996. *On-line stochastic processes in data compression.* Ph.D. thesis, University of Washington.

Burrows, M. & Wheeler, D.J. 1994. "A block-sorting lossless data compression algorithm." Technical report, Digital Equipment Corporation, Palo Alto, California.

Cleary, J.G. and Witten, I.H. 1984. "Data compression using adaptive coding and partial string matching." *IEEE Transactions on Communications*, **32**(4), 396–402.

Cleary, J.G., Teahan, W.J. and Witten, I.H. 1995. "Unbounded length contexts for PPM." *Proceedings DCC'95*, IEEE Computer Society Press.

Cleary, J.G. and Teahan, W.J. 1998. "Unbounded length contexts for PPM." *Computer Journal.* In press.

Francis, W.N. and Kučera, H. 1982. *Frequency analysis of English usage: Lexicon and grammar.* Houghton Mifflin, Boston.

Jelinek, F. 1990. "Self-organized language modeling for speech recognition," in A. Waibel and K. Lee, editors, *Readings in speech recognition*, 450–506. Morgan Kaufmann Pub.

Johansson, S., Atwell, E., Garside, R., and Leech, G. 1986. *The tagged LOB Corpus.* Norwegian Computing Centre, Bergen.

Kuhn, R. and De Mori, R. 1990. "A cache-based natural language model for speech recognition." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **12**(6), 570–583.

Malone, D. 1977. *Jefferson and his time.* Little Brown and Co., Boston.

Moffat, A. 1989. "Word based text compression." *Software—Practice and Experience*, **19**(2), 185–198.

Shannon, C.E. 1951. "Prediction and entropy of printed English." *Bell System Technical Journal*, 50–64.

Teahan, W.J. 1997. *Modelling English text.* D.Phil. thesis, Univ. of Waikato, N.Z.

Teahan, W.J. and Cleary, J.G. 1996. "The entropy of English using PPM-based models." *Proceedings DCC'96*, IEEE Computer Society Press.

Teahan, W.J. and Cleary, J.G. 1997. "Models of English text." *Proceedings DCC'97*, IEEE Computer Society Press.