Experiments on Sentence Boundary Detection

Mark Stevenson and Robert Gaizauskas
Department of Computer Science,
University of Sheffield
Regent Court, 211 Portobello Street,
Sheffield
S1 4DP United Kingdom
{marks, robertg}@dcs.shef.ac.uk

Abstract

This paper explores the problem of identifying sentence boundaries in the transcriptions produced by automatic speech recognition systems. An experiment which determines the level of human performance for this task is described as well as a memory-based computational approach to the problem.

1 The Problem

This paper addresses the problem of identifying sentence boundaries in the transcriptions produced by automatic speech recognition (ASR) systems. This is unusual in the field of text processing which has generally dealt with well-punctuated text: some of the most commonly used texts in NLP are machine readable versions of highly edited documents such as newspaper articles or novels. However, there are many types of text which are not so-edited and the example which we concentrate on in this paper is the output from ASR systems. These differ from the sort of texts normally used in NLP in a number of ways; the text is generally in single case (usually upper), unpunctuated and may contain transcription errors.[1] Figure 1 compares a short text in the format which would be produced by an ASR system with a fully punctuated version which includes case information. For the remainder of this paper error-free texts such as newspaper articles or novels shall be referred to as "standard text" and the output from a speech recognition system as "ASR text".

There are many possible situations in which an NLP system may be required to process ASR text. The most obvious examples are NLP systems which take speech input (eg. Moore et al. (1997)). Also, dictation software programs do not punctuate or capitalise their output but, if this information could be added to ASR text, the results would be far more usable. One of the most important pieces of information which is not available in ASR output is sentence boundary information. However, knowledge of sentence boundaries is required by many NLP technologies. Part of speech taggers typically require input in the format of a single sentence per line (for example Brill's tagger (Brill, 1992)) and parsers generally aim to produce a tree spanning each sentence. Only the most trivial linguistic analysis can be carried out on text which is not split into sentences.

```
GOOD EVENING GIANNI VERSACE ONE OF THE
WORLDS LEADING FASHION DESIGNERS HAS
BEEN MURDERED IN MIAMI POLICE SAY IT WAS
A PLANNED KILLING CARRIED OUT LIKE AN
EXECUTION SCHOOLS INSPECTIONS ARE GOING
TO BE TOUGHER TO FORCE BAD TEACHERS OUT
AND THE FOUR THOUSAND COUPLES WHO SHARED
THE QUEENS GOLDEN DAY
```

```
Good evening. Gianni Versace, one of
the world's leading fashion designers,
has been murdered in Miami. Police say
it was a planned killing carried out
like an execution. Schools inspections
are going to be tougher to force bad
teachers out. And the four thousand
couples who shared the Queen's golden
day.
```

Figure 1: Example text shown in standard and ASR format

It is worth mentioning that not all transcribed speech can be sensibly divided into sentences. It has been argued by Gotoh and Renals (2000) that the main unit in spoken language is the phrase rather than the sentence. However, there are situations in which it is appropriate to consider spoken language to be made up from sentences. One example is broadcast news: radio and television news programs. The DARPA HUB4 broadcast news evaluation (Chinchor et al., 1998) focussed on information extraction from ASR text from news programs. Although news programs are scripted there are often deviations from the script and they cannot be relied upon as accurate transcriptions of the news

[1] Speech recognition systems are often evaluated in terms of word error rate (WER), the percentage of tokens which are wrongly transcribed. For large vocabulary tasks and speaker-independent systems, WER varies between 7% and 50%, depending upon the quality of the recording being recognised. See, e.g., Cole (1996).

program. The spoken portion of the British National Corpus (Burnard, 1995) contains 10 million words and was manually marked with sentence boundaries. A technology which identifies sentence boundaries could be used to speed up the process of creating any future corpus of this type.

It is important to distinguish the problem just mentioned and another problem sometimes called "sentence splitting". This problem aims to identify sentence boundaries in standard text but since this includes punctuation the problem is effectively reduced to deciding which of the symbols which potentially denote sentence boundaries (.,!,?) actually do. This problem is not trivial since these punctuation symbols do not always occur at the end of sentences. For example in the sentence "`Dr. Jones lectures at U.C.L.A.`" only the final full stop denotes the end of a sentence. For the sake of clarity we shall refer to the process of discovering sentence boundaries in standard punctuated text as "punctuation disambiguation" and that of finding them in unpunctuated ASR text as "sentence boundary detection".

2 Related Work

Despite the potential application of technology which can carry out the sentence boundary detection task, there has been little research into the area. However, there has been work in the related field of punctuation disambiguation. Palmer and Hearst (1994) applied a neural network to the problem. They used the Brown Corpus for training and evaluation, noting that 90% of the full stops in this text indicate sentence boundaries. They used the part of speech information for the words surrounding a punctuation symbol as the input to a feed-forward neural network. But, as we mentioned, most part of speech taggers require sentence boundaries to be pre-determined and this potential circularity is avoided by using the prior probabilities for each token, determined from the Brown corpus markup. The network was trained on 573 potential sentence ending marks from the Wall Street Journal and tested on 27,294 items from the same corpus. 98.5% of punctuation marks were correctly disambiguated.

Reynar and Ratnaparkhi (1997) applied a maximum entropy approach to the problem. Their system considered only the first word to the left and right of any potential sentence boundary and claimed that examining wider context did not help. For both these words the prefix, suffix, presence of particular characters in the prefix or suffix, whether the candidate is honorific (Mr., Dr. etc.) and whether the candidate is a corporate designator (eg. Corp.) are features that are considered. This system was tested on the same corpus as Palmer and Hearst's system and correctly identified 98.8% of sentence boundaries. Mikheev (1998) optimised this approach and evaluated it on the same test corpus. An accuracy of 99.2477% was reported, to our knowledge this is the highest quoted result for this test set.

These three systems achieve very high results for the punctuation disambiguation task. It would seem, then, that this problem has largely been solved. However, it is not clear that these techniques will be as successful for ASR text. We now go on to describe a system which attempts a task similar to sentence boundary detection of ASR text.

Beeferman et al. (1998) produced a system, "CYBERPUNC", which added intra-sentence punctuation (i.e. commas) to the output of an ASR system. They mention that the comma is the most frequently used punctuation symbol and its correct insertion can make a text far more legible. CYBERPUNC operated by augmenting a standard trigram speech recognition model with information about commas; it accesses only lexical information. CYBERPUNC was tested by separating the trigram model from the ASR system and applying it to 2,317 sentences from the Wall Street Journal. The system achieved a precision of 75.6% and recall of 65.6% compared against the original punctuation in the text.[2] A further qualitative evaluation was carried out using 100 randomly-drawn output sentences from the system and 100 from the Wall Street Journal. Six human judges blindly marked each sentence as either acceptable or unacceptable. It was found that the Penn TreeBank sentences were 86% correct and the system output 66% correct. It is interesting that the human judges do not agree completely on the acceptability of many sentences from the Wall Street Journal.

In the next section we go on to describe experiments which quantify the level of agreement that can be expected when humans carry out sentence boundary detection. Section 4 goes on to describe a computational approach to the problem.

3 Determining Human Ability

Beeferman et. al.'s experiments demonstrated that humans do not always agree on the acceptability of comma insertion and therefore it may be useful to determine how often they agree on the placing of sentence boundaries. To do this we carried out experiments using transcriptions of news programmes, specifically the transcriptions of two editions of the

[2]Precision and recall are complementary evaluation metrics commonly used in Information Retrieval (van Rijsbergen, 1979). In this case precision is the percentage of commas proposed by the system which are correct while recall is the percentage of the commas occurring in the test corpus which the system identified.

BBC television program "The Nine O'Clock News".[3] The transcriptions consisted of punctuated mixed case text with sentences boundaries marked using a reserved character (";"). These texts were produced by trained transcribers listening to the original program broadcast.

Six experimental subjects were recruited. All subjects were educated to at least Bachelor's degree level and are either native English speakers or fluent second language speakers. Each subject was presented with the same text from which the sentence boundaries had been removed. The texts were transcriptions of two editions of the news program from 1997, containing 534 sentences and represented around 50 minutes of broadcast news. The subjects were randomly split into two groups. The subjects in the first group (subjects 1-3) were presented with the text stripped of punctuation and converted to upper case. This text simulated ASR text with no errors in the transcription. The remaining three subjects (4-6) were presented with the same text with punctuation removed but case information retained (i.e. mixed case text). This simulated unpunctuated standard text. All subjects were asked to add sentence boundaries to the text whenever they thought they occurred.

The process of determining human ability at some linguistic task is generally made difficult by the lack of an appropriate reference. Often all we have to compare one person's judgement with is that of another. For example, there have been attempts to determine the level of performance which can be expected when humans perform word sense disambiguation (Fellbaum et al., 1998) but these have simply compared some human judgements against others with one being chosen as the "expert". We have already seen, in Section 2, that there is a significant degree of human disagreement over the acceptability of intra-sentential punctuation. The human transcribers of the "Nine O'Clock News" have access to the original news story which contains more information than just the transcription. Under these conditions it is reasonable to consider their opinion as expert.

Table 1 shows the performance of the human subjects compared to the reference transcripts.[4]

An algorithm was implemented to provide a baseline tagging of the text. The average length of sentences in our text is 19 words and the baseline algorithm randomly assigns a sentence break at each word boundary with a probability of $\frac{1}{19}$. The two annotators labelled "random" show the results when this algorithm is applied. This method produced a very low result in comparison to the expert annotation.

Annotator	Text	P	R	F
1	Upper	84	68	76
2	Upper	93	78	85
3	Upper	90	76	82
4	Mixed	97	90	94
5	Mixed	96	89	92
6	Mixed	97	67	79
Random	Upper	5	5	5
Random	Mixed	5	5	5

Table 1: Results from Human Annotation Experiment

The performance of the human annotators on the upper case text is quite significantly lower than the reported performance of the algorithms which performed punctuation disambiguation on standard text as described in Section 2. This suggests that the performance which may be obtained for this task may be lower than has been achieved for standard text.

Further insight into the task can be gained from determining the degree to which the subjects agreed. Carletta (1996) argues that the kappa statistic (κ) should be adopted to judge annotator consistency for classification tasks in the area of discourse and dialogue analysis. It is worth noting that the problem of sentence boundary detection presented so far in this paper has been formulated as a classification task in which each token boundary has to be classified as either being a sentence boundary or not. Carletta argues that several incompatible measures of annotator agreement have been used in discourse analysis, making comparison impossible. Her solution is to look to the field of content analysis, which has already experienced these problems, and adopt their solution of using the kappa statistic. This determines the difference between the observed agreement for a linguistic task and that which would be expected by chance. It is calculated according to formula 1, where $\Pr(A)$ is the proportion of times the annotators agree and $\Pr(E)$ the proportion which would be expected by chance. Detailed instructions on calculating these probabilities are described by Siegel and Castellan (1988).

$$\kappa = \frac{\Pr(A) - \Pr(E)}{1 - \Pr(E)} \quad (1)$$

The value of the kappa statistic ranges between 1 (perfect agreement) and 0 (the level which would be expected by chance). It has been claimed that content analysis researchers usually regard $\kappa > .8$ to demonstrate good reliability and $.67 < \kappa < .8$ al-

[3]This is a 25 minute long television news program broadcast in the United Kingdom on Monday to Friday evenings.

[4]F-measure (F) is a weighted harmonic combining precision (P) and recall (R) via the formula $\frac{2PR}{P+R}$.

lows tentative conclusions to be drawn (see Carletta (1996)).

We began to analyse the data by computing the kappa statistic for both sets of annotators. Among the two annotators who marked the mixed case (subjects 4 and 5) there was an observed kappa value of 0.98, while there was a measure of 0.91 for the three subjects who annotated the single case text. These values are high and suggest a strong level of agreement between the annotators. However, manual analysis of the annotated texts suggested that the subjects did not agree on many cases. We then added the texts annotated by the "random" annotation algorithm and calculated the new κ values. It was found that the mixed case test produced a kappa value of 0.92 and the upper case text 0.91. These values would still suggest a high level of agreement although the sentences produced by our random algorithm were nonsensical.

The problem seems to be that most word boundaries in a text are not sentence boundaries. Therefore we could compare the subjects' annotations who had not agreed on any sentence boundaries but find that they agreed most word boundaries were not sentence boundaries. The same problem will effect other standard measures of inter-annotator agreement such as the Cramer, Phi and Kendall coefficients (see Siegel and Castellan (1988)). Carletta mentions this problem, asking what the difference would be if the kappa statistic were computed across "clause boundaries, transcribed word boundaries, and transcribed phoneme boundaries" (Carletta, 1996, p. 252) rather than the sentence boundaries she suggested. It seems likely that more meaningful κ values would be obtained if we restricted to the boundaries between clauses rather than all token boundaries. However, it is difficult to imagine how clauses could be identified without parsing and most parsers require part of speech tagged input text. But, as we already mentioned, part of speech taggers often require input text split into sentences. Consequently, there is a lack of available systems for splitting ASR text into grammatical clauses.

4 A Computational Approach to Sentence Boundary Detection

The remainder of this paper describes an implemented program which attempts sentence boundary detection. The approach is based around the `Timbl` memory-based learning algorithm (Daelemans et al., 1999) which we previously found to be very successful when applied to the word sense disambiguation problem (Stevenson and Wilks, 1999).

Memory-based learning, also known as case-based and lazy learning, operates by memorising a set of training examples and categorising new cases by assigning them the class of the most similar learned example. We apply this methodology to the sentence boundary detection task by presenting `Timbl` with examples of word boundaries from a training text, each of which is categorised as either `sentence_boundary` or `no_boundary`. Unseen examples are then compared and categorised with the class of the most similar example. We shall not discuss the method by which `Timbl` determines the most similar training example which is described by Daelemans et al. (1999).

Following the work done on punctuation disambiguation and that of Beeferman et. al. on comma insertion (Section 2), we used the Wall Street Journal text for this experiment. These texts are reliably part of speech tagged[5] and sentence boundaries can be easily derived from the corpus. This text was initially altered so as to remove all punctuation and map all characters into upper case. 90% of the corpus, containing 965 sentence breaks, was used as a training corpus with the remainder, which contained 107 sentence breaks, being held-back as unseen test data. The first stage was to extract some statistics from the training corpus. We examined the training corpus and computed, for each word in the text, the probability that it started a sentence and the probability that it ended a sentence. In addition, for each part of speech tag we also computed the probability that it is assigned to the first word in a sentence and the probability that it is assigned to the last word.[6] Each word boundary in the corpus was translated to a feature-vector representation consisting of 13 elements, shown in Table 2. Vectors in the test corpus are in a similar format, the difference being that the classification (feature 13) is not included.

The results obtained are shown in the top row of Table 3. Both precision and recall are quite promising under these conditions. However, this text is different from ASR text in one important way: the text is mixed case. The experimented was repeated with capitalisation information removed; that is, features 6 and 12 were removed from the feature-vectors. The results form this experiment are shown in the bottom row of Table 3. It can be seen that the recorded performance is far lower when capitalisation information is not used, indicating that this is an important feature for the task.

These experiments have shown that it is much easier to add sentence boundary information to mixed case test, which is essentially standard text with punctuation removed, than ASR text, even as-

[5] Applying a priori tag probability distributions could have been used rather than the tagging in the corpus as such reliable annotations may not be available for the output of an ASR system. Thus, the current experiments should be viewed as making an optimistic assumption.

[6] We attempted to smooth these probabilities using Good-Turing frequency estimation (Gale and Sampson, 1996) but found that it had no effect on the final results.

Position	Feature
1	Preceding word
2	Probability preceding word ends a sentence
3	Part of speech tag assigned to preceding word
4	Probability that part of speech tag (feature 3) is assigned to last word in a sentence
5	Flag indicating whether preceding word is a stop word
6	Flag indicating whether preceding word is capitalised
7	Following word
8	Probability following word begins a sentence
9	Part of speech tag assigned to following word
10	Probability that part of speech (feature 9) is assigned to first word in a sentence
11	Flag indicating whether following word is a stop word
12	Flag indicating whether following word is capitalised word
13	`sentence_boundary` or `no_boundary`

Table 2: Features used in `Timbl` representation

Case information	P	R	F
Applied	78	75	76
Not applied	36	35	35

Table 3: Results of the sentence boundary detection program

suming a zero word error rate. This result is in agreement with the results from the human annotation experiments described in Section 3. However, there is a far greater difference between the automatic system's performance on standard and ASR text than the human annotators.

Reynar and Ratnaparkhi (1997) (Section 2) argued that a context of one word either side is sufficient for the punctuation disambiguation problem. However, the results of our system suggest that this may be insufficient for the sentence boundary detection problem even assuming reliable part of speech tags (cf note 5).

These experiments do not make use of prosodic information which may be included as part of the ASR output. Such information includes pause length, pre-pausal lengthening and pitch declination. If this information was made available in the form of extra features to a machine learning algorithm then it is possible that the results will improve.

5 Conclusion

This paper has introduced the problem of sentence boundary detection on the text produced by an ASR system as an area of application for NLP technology.

An attempt was made to determine the level of human performance which could be expected for the task. It was found that there was a noticeable difference between the observed performance for mixed and upper case text. It was found that the kappa statistic, a commonly used method for calculating inter-annotator agreement, could not be applied directly in this situation.

A memory-based system for identifying sentence boundaries in ASR text was implemented. There was a noticeable difference when the same system was applied to text which included case information demonstrating that this is an important feature for the problem.

This paper does not propose to offer a solution to the sentence boundary detection problem for ASR transcripts. However, our aim has been to highlight the problem as one worthy of further exploration within the field of NLP and to establish some baselines (human and algorithmic) against which further work may be compared.

Acknowledgements

The authors would like to thank Steve Renals and Yoshihiko Gotoh for providing the data for human annotation experiments and for several useful conversations. They are also grateful to the following people who took part in the annotation experiment: Paul Clough, George Demetriou, Lisa Ferry, Michael Oakes and Andrea Setzer.

References

D. Beeferman, A. Berger, and J. Lafferty. 1998. CYBERPUNC: A lightweight punctuation annotation system for speech. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 689–692, Seattle, WA.

E. Brill. 1992. A simple rule-based part of speech tagger. In *Proceeding of the Third Conference on Applied Natural Language Processing (ANLP-92)*, pages 152–155, Trento, Italy.

L. Burnard, 1995. *Users Reference Guide for the British National Corpus.* Oxford University Computing Services.

J. Carletta. 1996. Assessing agreement on classification tasks: the kappa statistic. *Computational Linguistics*, 22(2):249–254.

N. Chinchor, P. Robinson, and E. Brown. 1998. HUB-4 Named Entity Task Definition (version 4.8). Technical report, SAIC. `http://www.nist.gov/speech/hub4_98`.

R. Cole, editor. 1996. *Survey of the State of the Art in Human Language Technology.* Available at: http://cslu.cse.ogi.edu/HLTsurvey/HLTsurvey.html. Site visited 17/11/99.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 1999. TiMBL: Tilburg memory based learner version 2.0, reference guide. Technical report, ILK Technical Report 98-03. ILK Reference Report 99-01, Available from `http://ilk.kub.nl/~ilk/papers/ilk9901.ps.gz`.

C. Fellbaum, J. Grabowski, S. Landes, and A. Baumann. 1998. Matching words to senses in WordNet: Naive vs. expert differentiation of senses. In C. Fellbaum, editor, *WordNet: An electronic lexical database and some applications.* MIT Press, Cambridge, MA.

W. Gale and G. Sampson. 1996. Good-Turing frequency estimation without tears. *Journal of Quantitave Linguistics*, 2(3):217–37.

Y. Gotoh and S. Renals. 2000. Information extraction from broadcast news. *Philosophical Transactions of the Royal Society of London, series A: Mathematical, Physical and Engineering Sciences.* (to appear).

A. Mikheev. 1998. Feature lattices for maximum entropy modelling. In *Proceedings of the 36th Meeting of the Association for Computational Linguistics (COLING-ACL-98)*, pages 848–854, Montreal, Canada.

R. Moore, J. Dowding, H. Bratt, J. Gawron, Y. Gorfu, and A. Cheyer. 1997. CommandTalk: A Spoken-Language Interface to Battlefield Simulations. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 1–7, Washington, DC.

D. Palmer and M. Hearst. 1994. Adaptive sentence boundary disambiguation. In *Proceedings of the 1994 Conference on Applied Natural Language Processing*, pages 78–83, Stutgart, Germany.

J. Reynar and A. Ratnaparkhi. 1997. A maximum entropy approach to identifying sentence boundries. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 16–19, Washington, D.C.

S. Siegel and N. Castellan. 1988. *Nonparametric Statistics for the Behavioural Sciences.* McGraw-Hill, second edition.

M. Stevenson and Y. Wilks. 1999. Combining weak knowledge sources for sense disambiguation. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 884–889. Stockholm, Sweden.

C. van Rijsbergen. 1979. *Information Retrieval.* Butterworths, London.