

Adding Sentence Boundaries to Conversational Speech Transcriptions using Noisily Labelled Examples

Tetsuya Nasukawa+, Diwakar Punjani, Shourya Roy, L. Venkata Subramaniam, Hironori Takeuchi+

+IBM Tokyo Research Lab,
Tokyo, Japan

IBM India Research Lab,
Block I, Indian Institute of Technology,
New Delhi, India.

Abstract

This paper presents a technique for adding sentence boundaries to text obtained by Automatic Speech Recognition (ASR) of conversational speech audio. We show that starting with imprecise boundary information added by using only silence information from an ASR system, we can improve boundary detection using head and tail phrases. The main purpose for the insertion of sentence boundaries to ASR conversational text is to improve linguistic analysis, namely Information Extraction, for text mining systems that handle huge volumes of textual data and analyze trends and specific features of concepts described in document sets. Hence, we also show how the addition of boundaries improves two basic natural language processing tasks viz. POS label assignment and NP extraction.

1 Introduction

Given conversational data in the form of audio files, the output of an Automatic Speech Recognition (ASR) System is a stream of words. However many applications such as information retrieval and natural language processing benefit from (or even require) a sentence structure. The inherent word recognition errors of an ASR system and presence of noise such as repeats, false starts and filler words in conversational speech makes identifying structural information a more challenging task as compared to well-written text.

Many researchers have worked in the area of identifying structures in speech transcriptions. Most of this work is based on supervised techniques which require prosodic as well as lexical features. These techniques learn sophisticated models based on carefully annotated data. However, in many real life settings generating such a well-labelled training dataset is difficult to the extent of impossible. We draw our motivation from the domain of call centers. Most companies today operate call centers as they allow them to be in direct contact with their customers, the number of calls handled by a call center

can range from a few hundreds to tens of thousands depending on the scale of operation. An ASR system deployed in such a call center can potentially produce large volumes of data everyday in the form of speech transcripts. This data is valuable for doing analysis at many levels, e.g., to identify problems and issues associated with different products and services and also to evaluate agents. The analysis is done using text processing techniques such as Parts Of Speech (POS) tagging, parsing, information extraction, summarization, etc.; these techniques benefit greatly from presence of sentence boundary information in the text.

While working on a real-life dataset, from the IT help desk of a company, we observed that other than timing information and the recognized words, no other information is available. We made similar observations with different datasets from call centers. For each new dataset it is not possible to create a well-labelled training set. Can we live without this well labelled training set? In this paper we show that we can train a boundary detection system based on only the silence information provided by the ASR system. We evaluate our system to show its effectiveness in two ways:

1. By comparing the boundaries added by the system with those added manually.
2. By comparing the outputs of a POS tagger on data with and without boundaries.

2 Importance for Information Extraction

The main purpose for the insertion of sentence boundaries to ASR conversational text is to improve linguistic analysis, namely information extraction, for text mining systems that handle huge volumes of textual data and analyze trends and specific features of concepts described in document sets, rather than to improve readability for human beings. Since the state of the art linguistic analysis tools such as POS taggers and syntactic parsers are based on sentence segmentation, correct identification of input sentences for such tools is critical for the quality of their outputs. In order to capture appropriate concepts through information extraction that

is based on linguistic analysis, a sentence boundary should not be inserted in the middle of a phrase or a multiple expression that represents a specific concept as a whole. Thus, mis-insertion (typically too much insertion) of sentence boundaries leads to lower recall in the information extraction. On the other hand, lack of appropriate sentence boundaries leads to lower precision in the information extraction as it tends to extract more noise (typically, inappropriate phrases). For example, in the extraction of noun phrases from the following expression "ok let me get my password hang on a second ok," without any sentence boundary in the middle, parser will likely end up extracting "my password hang," and "a second ok."

We would like to capture typical Noun Phrases (NP) that express objects for analysis, and typical Verb Groups (VG) that express actions and changes. Since such concepts are usually domain dependent, it is fundamentally better to apply domain knowledge. In addition, when working with ASR conversational text, it is essential to deal with noise caused by incomplete utterances and errors in recognition. The tendency for incomplete utterances and the recognition error rate are also domain dependent because they depend on the task of conversation and the system environment for speech recognition. For example, in the case of a help desk for IT system and a medical emergency call center, the way the caller speaks will be very different and the performance of the speech recognition system will also differ.

3 Background and Related Work

A large body of previous work exists in sentence boundary detection for both broadcast speech transcriptions [Gotoh and Renals, 2000] [Liu *et al.*, 2005] [Shriberg *et al.*, 2000] and conversational speech transcriptions [Kim *et al.*, 2004] [Liu *et al.*, 2005] [Shriberg *et al.*, 2000]. Most techniques use a combination of lexical and prosodic features where a manually marked text collection is used as a training set. Many of them use a Hidden Markov Model (HMM) approach to model the lexical features using n-gram language models [Gotoh and Renals, 2000] [Shriberg *et al.*, 2000]. Additional features such as prosody, including silence, are modeled as observation likelihoods attached to the n-gram states of the HMM [Shriberg *et al.*, 2000]. More recently Liu *et al.* used maximum entropy [Liu *et al.*, 2004] and conditional random fields [Liu *et al.*, 2005] to model a combination of lexical and prosodic features to obtain good sentence boundary detection on both broadcast speech and conversational speech.

Text analysis, information extraction and knowledge mining require processing of the text by performing syntactic and semantic analysis [Nasukawa and Nagano, 2001]. Such systems require well formed sentences for the natural language processing modules to work. In call center's the analysis of customer-agent conversations is very important for extracting key actionable insights. While some work appears on using the noisy transcriptions directly without sentence boundaries [Roy and Subramaniam, 2006], availability of text with sentence boundaries will greatly enhance information extraction from conversational text.

Contributions of this work: The contribution of this paper

is two-fold. Firstly, it proposes a simple technique to identify sentence boundaries in ASR transcriptions almost without any manual supervision. Most of the existing systems require training data carefully annotated using a number of prosodic features following guidelines such as [Strassel, 2003]. We only use the pause information, which is relatively immune to transcription noise, available in the transcriber output to build a language model, which is then used for identifying the sentence boundaries in the transcripts. The language model that we propose, as an alternative to the finite state models, is built using the opening and closing N-gram sequences of sentence units identified using the silence information in the training data. The time required for generating our model and using it to mark sentence boundaries is much less as compared to other models.

Secondly, since the value of the boundary detection is enhanced information extraction, we show results on how the addition of boundaries improves POS label assignment and NP and VG extraction. We also evaluated the performance of the proposed system on manually marked data and show that it works equally well in that setting.

4 Datasets

Before going into the details of the proposed technique let us look at some of the transcription datasets. The following section contains details of two manually transcribed and one automatically transcribed dataset.

4.1 Nature of Data

The first manually transcribed dataset we used is the *Switchboard Cellular Part 1 Transcription*, produced by the Linguistic Data Consortium (LDC)¹. This release contains 250 transcriptions of 5-6 minute telephonic conversations on various topics balanced by gender, under varied environmental conditions. These calls are transcribed using conventions similar to HUB-5 English [Hain *et al.*, 2000].

The voices of two speakers are recorded on two channels viz. *local channel* and *remote channel*. A *turn* has a speaker channel identification, and has a beginning and an end time stamp. The time stamp has both a start and an end point, and neither point can overlap a previous time stamp of the same speaker. The insertion of *breakpoints* has the same appearance as a new speaker turn. Breakpoints can be inserted wherever they seem convenient to the transcriber. They should occur at the natural boundaries of speech, such as pauses, breaths, etc. The following punctuation marks are used in the transcripts. The punctuation marks are primarily for ease of (human) reading.

- *periods* (.) are added at the end of declarative sentences
- *question marks* (?) are added at the end of interrogative sentences
- *commas* (,) are added between clauses as is accepted in the standard orthography of the language

The other manually transcribed dataset we used is also from LDC and known as *CallHome English Corpus*. It con-

¹<http://www ldc.upenn.edu/>

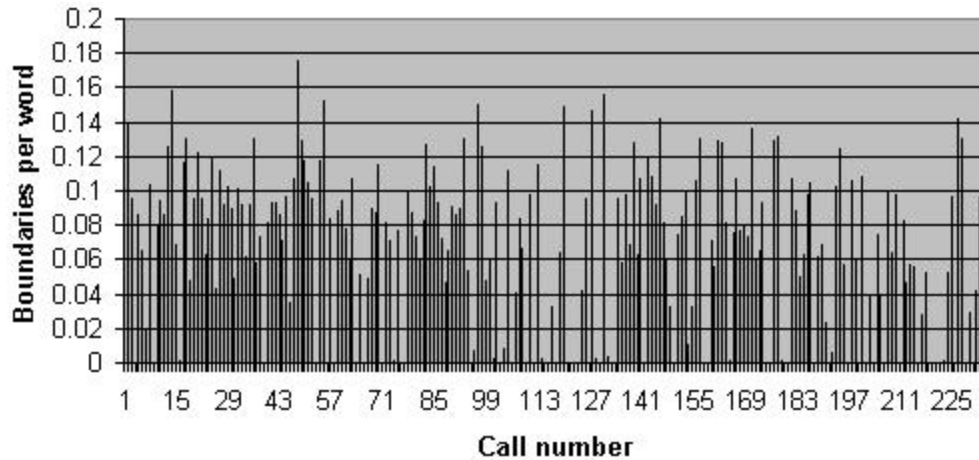


Figure 1: Boundaries per word for the switchboard dataset

sists of 120 unscripted telephone conversations between native speakers of English. The transcription scheme used was very similar to the scheme used for the Switchboard corpus and hence we are not mentioning it in detail.

The automatically transcribed dataset is from the technical *Helpdesk* of a large corporation. The ASR system was trained on approximately 300 hours of 6 KHz, mono audio data. The transcriptions contain *speaker identity*, *beginning time* and *duration of each word and actual word*. Special symbols such as *<s>*, *</s>* and, *SIL* are present in the transcriptions to indicate silence along with their durations.

Table 1 shows the summary of the three datasets. In this

Table 1: Summary of Data

	Swichboard	CallHome	Helpdesk
Total Calls	245	120	1720
Turns	37822	28633	56000
Words	266476	219846	1636889
Complete Turns	16597	10345	6429
Total Boundaries	20821	13886	92839

table, a *Complete Turn* is a turn that ends with a sentence boundary and *Total Boundaries* are the number of *'.'* and *'?'*.

4.2 Automatic Generation of Training Data

In the case of ASR transcribed data, presence of pause or silence in conversation is an indication of a sentence boundary. However, owing to the presence of spontaneity, hesitation, repetition, interruption in conversation, boundaries marked using only silence information is not accurate enough to be useful. It results in the marking of false boundaries. A lot of boundaries are missed because people also do not pause appropriately between sentence units. However, it is possible to build a *Language Model* from text with boundaries *noisily* marked using silence information and then use the *Language Model* to remove false boundaries and add the boundaries missed by the silence model. This simple method has

the advantage of being independent of carefully and manually created training data. We put a boundary when two or more consecutive silence characters (Sec 4.1) occur in the transcripts. Single silences are ignored since they represent very small pause duration. The boundaries so marked are then used for generating the language model.

For the manually transcribed datasets, the boundaries put by human transcribers are used for training the language model. We observed a distinct lack of consistency in marking boundaries in this case. The number of boundaries present should depend on the length of transcription (say *number of words*) and ideally *number of sentences marked* should vary *almost* proportionately to the *number of words* in the transcription. However, this is not true with both the manual datasets and there is a lot of subjectivity among transcribers. Figure 1 shows the high variation in average number of boundaries per word across different calls. This has a detrimental effect on any system trained on it.

5 Boundary Detection Using Head Phrases and Tail Phrases

In this section we will be talking about using imprecisely marked boundaries based on silence information to identify sentence boundaries in test data. It includes learning probable *head* and *tail* phrases and using them to mark sentence boundaries. A phrase which occurs at the beginning (or end) of a sentence is called a *head* (or *tail*) phrase respectively. Our technique is based on the observation there are some phrases which are more likely to be head (or tail) phrases than others. For example, phrases such as *hello this is*, *would you like* are typically found at the beginning of a sentence whereas phrases such as *help you today*, *click on ok* are commonly found at the end of a sentence. We describe our technique as a sequence of the following steps:

- **Data Cleansing:** Both manual as well as automatic transcription data require preprocessing. The objective of this step is to remove various transcription symbols

as well as noise introduced in the telephonic conversations due to their spontaneous nature. In presence of such noise, same head and tail phrases will appear corrupted and thereby reduce the statistical significance of the phrases. Cleaning in the case of manual transcriptions includes removing words which the transcriptionist has marked as unrecognizable. All proper nouns are replaced by the tag <PN> and interjections such as *mhm*, *uh-huh* are replaced by the tag <INT>. In the case of automatic transcriptions we only replace interjections by the tag <INT>. We observed that in the case of transcriptions of conversational data, cleansing or feature engineering makes a significant difference in the quality of results.

- **Identifying Head-Tail Phrases:** In Sec 4.2 we mentioned that the training data is annotated with boundaries marked based on silence information. We make a pass over the training data to identify all the head and tail phrases along with their frequencies. Any phrase of length k which appears at the beginning or end of the sentence is selected as head or tail phrase respectively. The value of k depends on the exact dataset. In our case, we use $k=3$ for all the experiments. This initial list of phrases is pruned based on following thresholds:

1. The *support* or *total frequency* of each phrase should be greater than a threshold t_1 . This is to minimise the number of spurious head and tail phrases. We used t_1 in the range of 3 to 5.
2. Some phrases appear in the beginning (or end) as well as in the middle of sentences almost equally. Example of such phrases could be *I dont know*, *thanks for calling*. We need to penalize such phrases in comparison to phrases which typically appear as head or tail phrases. For each identified head phrase (or tail phrase) we compute *goodness score* (s_h) (or s_t) as following.

$$s_h = \frac{\#Occurrences\ as\ Head\ Phrase}{\#Total\ Occurrences}$$

$$s_t = \frac{\#Occurrences\ as\ Tail\ Phrase}{\#Total\ Occurrences}$$

It can be readily seen that $0 < s_h, s_t \leq 1$. Good head or tail phrases should have s_h and s_t scores close to 1.

3. Owing primarily to the spontaneous nature of the conversations and the noise in the data we have observed that people do not complete their sentences often. This may happen because of different speaker behaviours viz. repair and restart. A *repair* happens when the speaker attempts to correct a mistake that he or she just made. In a *restart*, the speaker abandons a current utterance completely and starts a new one. As a result, we observed that tail phrases are comparatively less reliable for detecting sentence boundaries than head phrases. Hence, we prune the list of tail phrases (based on s_t values) significantly compared to head phrases.

For example, in the case of helpdesk dataset, we kept 2984 head phrases and 510 tail phrases.

4. To combat the noise introduced by the ASR system, we further pruned the list of head and tail phrases based on two lists of *impermissible headwords* and *impermissible tailwords*. These lists contain words which can not appear at the beginning of a head phrase and at the end of a tail phrase respectively. These are primarily *conjunctions*, *prepositions* etc.
- **Inserting Sentence Boundaries** The selected head and tail phrases are used for annotating the call transcripts. A boundary is marked before every head phrase and after every tail phrase. If turn change information is available then that is also treated as an end of a sentence unit unless the turn ended with a phrase in the *impermissible tailwords* list. If a phrase occurs both as a head phrase and a tail phrase it is considered to be a head phrase as discussed in point 4 above. Even though we have a fixed phrase length (k), no minimum sentence length is set due to the nature of data, a lot of one or two word sentence units are present.
 - **Removing False Boundaries** As discussed in section 4.2 using only silence duration for marking sentence boundaries results in a lot of false boundaries. In this step we use the *goodness score* (s_h and s_t) for removing the incorrect boundaries marked using the silence information. At the boundaries marked using silence information only the *goodness scores* are looked up for the head and tail phrases and if either of the values is below a threshold then the boundary is considered to be incorrect. A sentence boundary put following words from the *impermissible tailwords* list has also been observed to be typically incorrect and hence is removed.

6 Evaluation

In this section, we present the evaluation of our punctuation insertion method. We also look at how POS tagging and NP extraction are affected by the presence or absence of sentence boundaries.

6.1 Experiments

In these experiments, we evaluate the performance of our punctuation insertion method. For training, we use the two type of datasets mentioned in Sec 4.1, manual transcriptions and automatic transcriptions. The manual transcriptions used are the Swichboard corpus and the CallHome corpus from Linguistic Data Consortium (LDC). These are transcribed phone conversations where punctuations are manually inserted. The automatic transcription dataset used is the Helpdesk data. In this data, punctuations are automatically inserted based on silence.

As the training data, we use 210, 100 and 1600 calls from Swichboard, CallHome and Helpdesk corpus respectively.

For the evaluation of our punctuation insertion method on the manual transcriptions, we use 35 and 20 calls from Swichboard and CallHome corpus as the test data. Table 2 shows the evaluation results for these datasets.

Table 2: Result of Punctuation Insertion on Manual transcriptions

	Precision	Recall	F1	WER
Switchboard	0.55	0.78	0.65	0.78
Call-home	0.52	0.68	0.59	0.95

In this table, F1 denotes the F1-measure, which is the harmonic mean of traditional precision and recall measures and Word Error Rate (WER) is the ratio of total number of erroneous punctuations inserted to the total number of punctuations in the test data. As we have mentioned in Section 4.2, in the manual datasets the boundaries are marked very subjectively. Hence, our actual performance numbers look better than the numbers shown in Table 2. In fact, removing calls with very few boundaries increases our F-score by almost 10%.

The system was evaluated on 20 manually labelled calls in case of the helpdesk corpus, the results for silence model, language model and a combination of the two models are presented in Table 3. As we can see in the table the silence model has a very low recall which results in a F1 score of only 0.37 and a WER of nearly 1. Our language model has significantly better precision and recall as compared to the silence model; it has 0.65 F1 score and a WER of only 0.60. The combined model has an improved recall with respect to the language model but there is a decrease in precision. The best results are achieved when false boundaries are removed from the output of the combined model resulting in a F1 score of 0.70 and 0.58 word error rate.

Table 3: Result of Punctuation Insertion for Helpdesk data

	Precision	Recall	F1	WER
Baseline: Silence only	0.54	0.28	0.37	0.96
HT only	0.78	0.55	0.65	0.60
HT+Silence	0.66	0.72	0.68	0.66
HT+Silence-FB	0.72	0.69	0.70	0.58

In this table *HT* refers to the Head phrase and Tail phrase method of this paper. *HT + Silence* refers to putting boundaries using both HT and silence information. *HT + Silence - FB* refers to removing False Boundaries as described in Section 5.

6.2 Evaluation in an IE Task

In many text mining systems, we extract useful information from text using Natural Language Processing (NLP) techniques. Since the state of the art linguistic analysis tools such as POS taggers and syntactic parsers are based on sentence segmentation, correct identification of input sentences for such tools is critical for the quality of their outputs. We study the impact of punctuation insertion on the results of some linguistic analysis tools. In particular we look at Part Of Speech (POS) tagging and Noun Phrase (NP) extraction.

In our first evaluation, we compare the results of POS tagging on text without punctuations and with punctuations added by our method. We use the manually added POS tags as the gold standard data and calculate accuracies by comparing with it.

Table 4: Accuracy of POS tagging (Switchboard)

	None	System
Acc.	0.9204	0.9542

Table 5: Accuracy of POS tagging (Helpdesk)

	None	Silence	HT	Silence+HT	Silence+HT-FB
Acc.	0.9493	0.9493	0.9663	0.9656	0.9680

Table 4 shows accuracies of POS tagging on two text data from Switchboard corpus, data without punctuations (None) and data with punctuation by our method (System). From this result, it is found that POS tagging is improved by adding punctuations. This difference is meaningful because t-test rejects the possibility of an accidental difference between them at a probability level of $\alpha = 0.01$. Table 5 shows accuracies of POS tagging on 20 calls from Helpdesk data. We can see that our *HT+Silence-FB* method gets best performance. In these results, the differences between *Silence* and *HT* and between *Silence+HT* and *Silence+HT-FB* are meaningful under the t-test at a probability level of $\alpha = 0.01$.

Table 6 and 7 show the results of the POS tagging for frequent 10 keywords in 35 calls from Switchboard data set and 121 calls from Helpdesk data set. In Switchboard dataset, we also have the case where punctuations are inserted manually. In this data set, to the data without punctuation (None), data with punctuation by our method (System) and data with manual punctuation (Manual), we apply the preprocess of our text mining system. For the Helpdesk dataset, to the data without punctuation (None), data with punctuation by our method (HT), data with punctuation based on silence (Silence), data with punctuation based on both silence and our method (Silence+HT) and data with false boundaries removal from punctuation based on Silence and HT method (Silence+HT-FB), we apply the preprocess of our text mining system. From these results, it is seen that the identification of not-noun keywords is improved by using our punctuation insertion method while keeping noun keyword identification capability intact. In the Switchboard dataset, "i" should be detected as pronoun and "yeah" and "oh" should be detected as interjection. Table 8 shows the detailed results for these words. It is found that incorrect POS labels are assigned to these three keywords much more frequently in the non-punctuation data. In conversational data, these pronoun and interjection keywords appear very frequently. If lots of improper POS tags are assigned to these keywords, the result of information extraction based on POS information tends to be incorrect. So, inserting punctuation information is very important in a text mining system that relies on POS information for information extraction from text. We believe our punctuation insertion method will improve the result of a text mining system for conversational data based on these results.

The difference between *Silence*, *HT*, *Silence+HT* and *FB* can be inferred to mean that the punctuation insertion based on our technique has an impact on POS tagging.

Table 9 is the ranking of POS information extracted from

Table 6: Result of POS tagging for frequent 10 keywords (Switchboard)

surface	None		System		Manual	
	noun	not-noun	noun	not-noun	noun	not-noun
i	1490	447	250	1687	481	1456
yeah	406	358	310	454	304	460
oh	144	240	166	218	162	222
am	119	63	23	159	38	144
work	55	75	36	94	41	89
thing	77	0	76	1	77	0
money	32	0	32	0	32	0
place	25	0	24	1	24	1
call	63	52	61	54	62	53
study	13	3	13	3	13	3

Table 7: Result of POS tagging for frequent 10 keywords (Helpdesk)

surface	None		Silence		HT		Silence+HT		Silence+HT-FB	
	noun	not-noun	noun	not-noun	noun	not-noun	noun	not-noun	noun	not-noun
right	291	1005	315	981	359	937	374	922	376	920
yeah	746	378	730	394	604	520	544	580	549	575
click	370	344	385	329	432	282	428	286	411	303
number	616	25	629	12	634	7	633	8	633	8
note	313	148	318	143	321	140	326	135	326	135
hold	121	90	111	100	93	118	88	123	87	124
let	145	423	129	439	110	458	104	464	106	462
password	288	1	288	1	288	1	288	1	288	1
look	42	284	38	288	39	287	39	287	40	286
ticket	372	4	372	4	372	4	372	4	371	5

Table 8: Detailed result of POS tagging (Switchboard)

surface	None				System				Manual			
	noun	pronoun	interjection	misc	noun	pronoun	interjection	misc	noun	pronoun	interjection	misc
i	1490	58	0	389	250	1640	0	47	481	1355	0	101
yeah	406	0	1	357	310	0	360	94	304	0	331	129
oh	144	0	28	212	166	0	174	44	162	0	142	80

Table 9: Top 10 POS information with frequency in the data set (Helpdesk)

surface	None		Silence		HT		Silence+HT		Silence+HT-FB	
	verb	noun	verb	noun	verb	noun	verb	noun	verb	noun
verb	30417		30141		30095		29959		29946	
noun	24451		24967		25276		25814		25824	
pronoun	20399		20389		20409		20400		20403	
adposition	12657		12639		12645		12627		12627	
determiner	11227		11352		11433		11500		11497	
adverb	10408		10421		10258		10257		10254	
conjunction	8529		8394		8339		8293		8294	
adjective	7360		6698		6376		6112		6116	
numeral	4834		4807		4831		4811		4821	
interjection	1666		2152		2327		2234		2223	

Table 10: Extracted Top 10 Noun Phrases (Switchboard)

None	System	Manual
a lot	a lot	a lot
i i	yeah yeah	yeah yeah
i am	a little bit	a little bit
yeah yeah	the computer	the computer
yeah i	the people	the people
no i	my god	my god
a little bit	i am	i am
iguess	the phone	i i
the people	a couple	a couple
that i	the union	the phone

Table 11: Extracted Top 10 Noun Phrases (Helpdesk small data)

None	Silence	HT	Silence+HT	Manual
all right	all right	all right	all right	all right
the password	the password	the password	the password	the password
a ticket	my computer	my computer	i'm gonna	i'm gonna
a problem	i'm gonna	i'm gonna	my computer	yeah ok
my computer	a ticket	a ticket number	your ticket number	my computer
i'm gonna	a problem	your ticket number	a ticket number	your ticket number
the printer	the printer	a problem	ok i'm	i'm sorry
an option	your ticket number	ok i'm	a problem	a ticket
ok ok	ok i'm	an option	yeah ok	a problem
right yeah	an option	windows thank	the printer	the printer

Table 12: Extracted Top 10 Noun Phrases (Helpdesk)

None	Silence	HT	Silence+HT	Silence+HT-FB
all right	all right	all right	all right	all right
a problem	a problem	a problem	a problem	a problem
the one	the one	the one	the one	the one
the password	the password	the password	the password	the password
a ticket	a minute	ok thank	a minute	a minute
a minute	the ticket number	a minute	lotus notes	lotus notes
ok ok	lotus notes	right click	right click	right click
the ticket number	a ticket number	ok ok	the ticket number	the ticket number
a lot	ok ok	lotus notes	ok thank	ok thank
lotus note	ticket	the ticket number	a ticket	a ticket

Helpdesk data set. From this table, it is seen that the boundary insertion has an effect on the POS labels. Table 10, 11 and 12 show extracted frequent 10 noun phrases in each data set. It can be seen that noun phrase extraction is improved by inserting punctuations. For example on the switchboard dataset the extracted NPs for manual boundaries and system boundaries are comparable and markedly improved compared to no boundaries.

7 Conclusions

Most current boundary detection techniques are based on training a system on a, carefully prepared, manually annotated boundaries dataset. In this paper we have shown that we can train a sentence boundary detector based on only the silence information provided by an ASR system. Such a sentence boundary detector is shown to have a good accuracy, and that it can be further improved by combining it with the silence model. The language model is also shown to perform competently on manually transcribed data. We also demonstrate how boundary detection results in improved POS label assignment and NP extraction, two key preprocessing steps in information extraction. As a future direction it would be interesting to measure the improvement due to boundary detection in an information extraction task.

References

- [Gotoh and Renals, 2000] Yoshihiko Gotoh and Steve Renals. Sentence boundary detection in broadcast speech transcripts. In *Proc. International Speech Communication Association (ISCA) Workshop: Automatic Speech Recognition: Challenges for the New Millennium (ASR-2000)*, Paris, France, September 2000.
- [Hain *et al.*, 2000] Thomas Hain, Philip Woodland, Gunnar Evermann, and Dan Povey. The cu-htk march 2000 hub5e transcription system, 2000.
- [Kim *et al.*, 2004] Joungbum Kim, Sarah E. Schwarm, and Mari Ostendorf. Detecting structural metadata with decision trees and transformation-based learning. In *HLT-NAACL*, pages 137–144, 2004.
- [Liu *et al.*, 2004] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper. Comparing and combining generative and posterior probability models: Some advances in sentence boundary detection in speech. In *Proc. of EMNLP*, pages 64–71, Barcelona, Spain, July 25-26 2004.
- [Liu *et al.*, 2005] Y. Liu, A. Stolcke, E. Shriberg, and M. Harper. Using conditional random fields for sentence boundary detection in speech. In *Proc. of ACL-05*, pages 451–458, Ann Arbor, MI, USA, June 25-30 2005.
- [Nasukawa and Nagano, 2001] T. Nasukawa and T. Nagano. Text analysis and knowledge mining system. *IBM Systems Journal*, pages 967–984, 2001.
- [Roy and Subramaniam, 2006] Shourya Roy and L. Venkata Subramaniam. Automatic generation of domain models for call centers from noisy transcriptions. In *Proc. of COLING/ACL 06*, pages 737–744, Sydney, Australia, July 2006.
- [Shriberg *et al.*, 2000] E. Shriberg, A. Stolcke, D. Hakkani-Tur, and G. Tur. Prosody-based automatic segmentation of speech into sentences and topics. *Speech Communication*, pages 127–154, 2000.
- [Strassel, 2003] Stephanie Strassel. Simple metadata annotation specification. Annotation guide, Linguistic Data Consortium, 2003. Version 5.0 – <http://www ldc.upenn.edu/Projects/MDE/>.