Variable Span Disfluency Detection in ASR Transcripts

Rahul Gupta[1]*, Sankaranarayanan Ananthakrishnan*[2]*, Zhaojun Yang*[1]*, Shrikanth Narayanan*[1]

[1] Signal Analysis and Interpretation Laboratory, University of Southern California
Los Angeles, CA 90081, U.S.A.
[2] Speech, Language, and Multimedia Business Unit, Raytheon BBN Technologies
Cambridge, MA 02138, U.S.A.

Abstract

Natural conversations often involve disfluencies in the form of revisions, repetitions, interjections, filled pauses and such. This paper focuses on word/phrase repetitions and revisions that are lexically well formed. These are generally captured by an ASR but pose problems to downstream processing such as spoken language translation (SLT). We describe a system to identify such word level disfluencies with a goal towards removing them in real time from the automatic recognition (ASR) system output. We use a span based training system to utilize the contextual information while tagging disfluencies. We design our system on the oracle transcripts and test them on both reference and ASR transcripts. We achieve an area under the receiver operating characteristics (ROC) curve for word level disfluency detection of .93 and .87 for the reference and the ASR transcripts respectively.

Index Terms: Disfluency detection, span based detection

1. Introduction

Disfluencies such as revisions, word repetitions, phrase repetitions, fillers (e.g. ah, um, eh etc.) are common in natural spoken language. Previous work on disfluency includes studies on structures in disfluency production [1, 2, 3], acoustics characteristics [4] and their effect on human language comprehension [5]. Disfluencies can be deemed extraneous units in an utterance. The removal of these disfluencies yields the fluent ancillary (the intended utterance), often desired by many spoken language technology applications including speech understanding and translation. There are specific types of disfluency such as word and phrase repetitions and revisions which are lexically well formed (and hence can be captured by an automatic speech recognition system). However their presence still can cause problems for processing downstream applications such as speech understanding and translation. The focus of this paper is the detection of word-level disfluencies in ASR transcripts using context based lexical and other ASR-derived features, with a view toward aiding downstream machine translation.

Johnson [6] categorizes speech disfluencies into eight categories, which include nonverbal vocalizations (e.g., fillers like um, ah), word fragments and full lexical items. Of these, we focus on detecting word repetitions, phrase repetitions, incomplete phrases and revisions in this paper. Our experiments are done using both oracle reference transcriptions and noisy ASR transcriptions. Comparison of results over these two sources helps us gauge the usefulness of designed features as those in one case may not be as informative as in the other. A few examples of such disfluencies are shown in Table 1. As can be seen, the removal of these disfluencies does not alter the intended information conveyed but instead helps clean up the surface text to be conducive to further automated processing. Note that this work of disfluency detection from text does not focus on fillers such as ah and um since they are trivial to mark automatically once detected from the audio stream. Further, partly spoken and broken words are removed from the database for the purpose of training the ASR system in our experiments and hence they do not occur in either the reference or the ASR transcripts by design.

Disfluency type	Example
Word repetition	**Where** Where are you going ?
Phrase repetition	So **can you** can you tell about their location ?
Revision	I think **there is bad ah** the security is really bad
Incomplete phrase	**His name** where does he live ?

Table 1: Examples of disfluencies

Previous studies have aimed at detecting disfluencies in human conversation given the true reference transcripts as well as ASR transcripts. Researchers [7, 8, 9] have focused on the removal of disfluencies for improving spoken language translation on true transcripts using word n-gram features. Prosodic features have also been used along with word level features for disfluency detection [10, 11, 12]. Howsoever, most of these schemes tag words individually using a local tagger (maximum entropy model) or sequential classifier (linear chain conditional random field) which only incorporates immediate context into account. Moreover, prosodic features are obtained from oracle sources or systems independent of ASR transcription. Lease et al. [13] use features based on parsing trees to identify filled pauses, discourse markers and explicit editing terms to incorporate syntactic structure into account. They provide results on ASR transcripts, however do not make use of prosodic information readily available from the ASR output.

We train a word span based classification system that tags spans of words instead of a single word, in contrast to previous models. As seen in the examples, some disfluencies do not occur as isolated words. This model helps us to tag a word span as disfluency at the same time incorporating longer context into account which is not possible with simple sequential modeling schemes. We then derive the word level predictions based on the outputs on such spans. Our baseline system to obtain the final predictions is based on a few simple lexical features. We further augment our system with features based on typed dependencies amongst words as well as the length of silence before each word to incorporate syntactic structure and prosodic information respectively. We compare the usefulness of these features across the two transcription sources and observe dissimilar performances. Our results suggest that whereas the performance in the case of ASR transcripts degrades considerably when just using the baseline features as compared to that of oracle transcripts, the additional features provide a greater discrimination. We obtain an area under ROC curve for detection of disfluencies at word level of .93 and .87 with reference and ASR transcripts, respectively.

2. Training and Evaluation Data

For the experiments of this paper, we use the English side of the DARPA TransTac two-way spoken dialog collections covering various domains, including force protection (e.g., checkpoint, reconnaissance, and patrol), medical diagnosis, aid, maintenance, infrastructure, and others [15]. We test our system on both the reference transcripts and ASR transcripts of this data. We use the BBN Byblos ASR system to transcribe the speech automatically. The system uses a multi-pass decoding strategy in which models of increasing complexity are used in successive passes to obtain gradually refined recognition hypotheses [16]. The acoustic model (AM) was trained on approximately 200 hours of manually transcribed conversational English speech, which consisted of 129K utterances segmented by sentence boundaries. We trained the language model (LM) on 6M sentences with 60M words, drawn from both the TransTac domain and other out-of-domain sources.

2.1. Training Corpus

We manually annotated a subset of the true reference transcriptions (11501 utterances) of the 200 hours of English speech with disfluency labels. Each of these utterances necessarily contains at least one disfluency. We train our disfluency detection models on this subset instead of the entire training set to maintain the class balance while training. The rest of the training data contains very few disfluencies, and therefore we do not include it in our training set. We train our model on the features derived from reference transcript as well as the length of silence before each word as output by the ASR system. We employed 10-fold jack-knifing technique to decode the training corpus to obtain the silence timings before each word. We divided the 200 hours of English speech with reference transcriptions into ten equal partitions. Each partition was decoded with an LM that left out transcriptions for that partition (ten different LMs were trained, one for each partition). We do this so that we do not over-fit the LM to obtain the silence features for the training set. The global baseline AM was used for decoding all partitions. We list the specifications of the training and evaluation set in Table 2.

We observe a considerably higher word error rate (WER) for the training set. This corroborates our earlier note that the presence of disfluencies causes more frequent errors in ASR systems. This is intuitive given the fact that the LMs are trained mainly on fluent data thus reducing the likelihood of an ASR hypothesis that contains disfluencies. Owing to the high WER, we train our models just on the reference transcripts and not on the ASR transcripts.

2.2. Evaluation Corpus

We use 1354 utterances as the development set and 1450 as the testing set. Again, we manually annotated these sets for disfluencies to obtain the ground truth labels. These sets have more natural distribution of disfluencies because we did not restrict them to only those utterances that contained disfluencies, as we did in training. After decoding both these sets, we obtain a better WER when compared to the training set as only 175 of development set utterances and 171 of testing set utterances contain disfluencies. We evaluate our disfluency detector both on the reference transcript as well as the transcripts obtained from the ASR system.

Corpus	Size (words)	Size (Disfluent words)	WER
Training	209k	21k	39.2
Dev.(ref. transcript)	17k	334	-
Dev.(ASR transcript)	17k	314	13.5
Testing(ref. transcript)	16k	347	-
Testing(ASR transcript)	16k	331	11.1

Table 2: Training and evaluation datasets

In order to obtain the ground truth label for ASR transcripts we map the labels from the reference transcripts using alignment. The labels for all the correctly aligned and replaced words in the reference transcripts are directly mapped to the corresponding words in the ASR hypotheses. All the insertions in the ASR transcripts were marked as not being a disfluency. As seen in Table 2, the number of disfluent words in the development set and the test set in case of ASR transcripts is fewer than in the reference transcripts (difference equals the number of deleted disfluent words).

3. Disfluency detection

Following the SPANMAXENT-ASR approach for named entity detection in [17], we use a variable span scheme for disfluency detection. As the disfluencies can range over several words, instead of tagging each word individually, it makes intuitive sense to tag an entire span of words as a disfluency. Also as the disfluencies disrupt the canonical syntactic structure of an utterance, local contextual information can help us identify irregularities corresponding to a disfluency. With this model, we are able to better capture contextual information in the utterance by treating spans as atomic units for labeling disfluencies. We next describe our training methodology using features on word spans followed by the inference methodology to obtain word level decisions.

3.1. Training methodology and features

In order to obtain the training instances, we use each individual span corresponding to a disfluent and non-disfluent region separately. All of our features are obtained from within the span with each word span containing words either from the disfluent or fluent region only. As an example, for the utterance shown in Table 1, “I think **there is bad ah** the security is really bad” there is one region corresponding to a disfluency and two regions which are not disfluent. We find all spans of consecutive words for each region, up to a specified maximum length and assign to each span the label of the containing region. The spans up to a length of 3 for this example are listed in Table 3. As mentioned in the introduction, in this work, we do not focus on detecting fillers as “ah”.

We train the model using features for each span. Additionally, we use context from the words immediately to the left/right of the span. For example all span internal words are considered as word identity features for the current span whereas the left and right words are added as lexical context. In the given example, if “there is bad” is the current span, (there, is, bad) are used as internal word identity features for the span while (think) and (the) are used as left and right context, respectively. Hence for a span length 3 and using one right/left word context will give us features for the span “there is bad” along with the features for the words “think” and “ the”. Again, if the right or left context is a filler we ignore it to obtain contextual features as they do not contain any lexical meaning and instead use the next closest neighbor.

Span length	Not-Dis.	Dis.	Not-Dis.
	I think	there is bad	ah the security is really bad
1	I, think	there, is, bad	the, security, is, really, bad
2	I_think	there_is, is_bad	the_security, security_is, is_really, ...
3	NA	there_is_bad	the_security_is, security_is_really, is_really_bad

Table 3: Example training instances of variable span lengths.

We set our baseline based on simple lexical features that are then augmented with typed dependency and timing features. The baseline features are solely designed to capture the repetitive nature of disfluencies as well as presence of specific words (as fillers) around disfluencies. The additional features capture both the structure of the whole utterance and the prosodic information about pauses detected by the ASR system. We trained a maxent classifier on features from each span length with the L-BFGS [18] algorithm and the Gaussian priors were tuned empirically to be 0.05 on the development set. Note we do not use sequential models (e.g. linear chain conditional random field model) as it is not trivial to train a sequential model particularly on multi-word spans. Also, simple sequence modeling schemes only account for immediate neighbors instead of a longer context.

3.1.1. Baseline features

• **Current word identity**: We use the current words in the span as features by themselves. This in itself is like training separate LMs for the fluent and disfluent regions. In case of a multi-word span, this feature will help the model determine if this conjunction of words belongs to a disfluency or not.

• **Filler in span**: This feature indicates the presence of a filler (ah, um etc.) in the current span of words. It is motivated by the observation that word-level disfluencies are often accompanied by fillers.

• **Filler before/after the span**: This feature indicates the presence of fillers right before or after the span of words. This is similar to context features but this feature just indicates their presence. We do not extract any other feature for fillers in context.

• **Word level string edit distance between current span and following span**: We calculate the string edit distance between the current span and the following words considering each word to be an element of the string. We compare the current span of length L to next $L-1$, L and $L+1$ words. This feature is expected to capture word repetitions, phrase repetitions and also revisions with some degree of reuse of words. We do not include fillers while calculating string edit distances.

• **Character level string edit distance between current span and the following span**: We perform the same comparison as above, but instead consider each character to be an element of the string while calculating the string edit distance. Apart from repetitions, this feature is expected to capture revisions with false starts and rephrasing the utterance.

• **Indicator representing use of a discourse marker**: Certain phrases such as “you know”, “like” help beginning or keeping a turn or serving an acknowledgment. However, we observe that they get marked as disfluency even when they are the part of intended utterance (e.g., do you know him?). We prepared a list of n-grams for the discourse markers corresponding to their meaningful usage. If we observe an n-gram around a discourse marker that is in the list, we set this feature to 1.

3.1.2. Additional features

Disfluencies with any kind of lexical variability, particularly revisions, are difficult to capture using above features. Additionally, errors may be further aggravated in case of ASR errors involving word replacements. Furthermore all word and phrase repetitions are marked as a disfluency using the above scheme of features even if they are actually not (e.g., “He said that that belongs to him” does not contain a disfluency). Also lexically similar words appearing close to each other tend to be marked as disfluency. Therefore we use the features discussed below to capture the structure of the sentence and the pausing behavior before each word for a better detection of disfluencies in case of higher lexical variability and errors introduced during ASR transcriptions.

• **Number of incoming and outgoing typed dependencies for each span**: We parse each utterance using the Stanford parser [19] and obtain typed dependencies between all the words in an utterance. We count the number of incoming and outgoing dependencies from each word in the span. We observe that the number of dependencies for disfluencies is smaller than the non-disfluent words. This is also intuitive as disfluencies are extraneous words in the utterance and do not fit the canonical syntactic flow. This feature is expected to work well even in the case of word replacements in ASR as they should have lower number of dependencies.

• **Length of silence before each word**: We also use the length of silence before each word as a feature since disfluencies are often accompanied by a pause around them. We obtain the length of silence from the ASR transcription. Since we evaluate the performance both on the reference and the ASR transcript, we map the silence phonemes as follows:

True transcript: We align the reference and the ASR transcripts. For each aligned word (correct or replacement) in the reference transcript, we map the silence length before the aligned ASR word as the corresponding silence length. For the remaining words, this feature is not available.

ASR transcript: We directly use the silence length before each hypothesized word.

We uniformly bin the silence lengths (bin length 0.1 seconds) to discretize this feature. We do not use other ASR based prosodic features because in case of an erroneous hypothesis, such features correspond to the erroneous word instead of the reference word. The use of this silence feature is motivated towards designing features readily available from ASR transcriptions. Such ASR based features are expected to be a better match for detection as compared to features from other sources.

3.2. Inference methodology

During inference, we do not have the labeled regions on the development and the test set. Therefore, we iteratively tag each span with the maxent classifier, up to a maximum span length and then aggregate the results. In the first pass, we tag all single-word spans; in the second pass, we tag all two-word spans; and so on. We obtain a posterior probability for each span being a disfluency during each pass. In the next step we calculate the probability of a word being a disfluency from these span probabilities. In order to decide if a word corresponds to a disfluency, we perform a weighted combination of classifier posteriors for all spans covering that candidate hypothesis word. For instance, the final score for the word “there” in the example for a model trained with maximum span length 3 is computed as shown in equation 1.

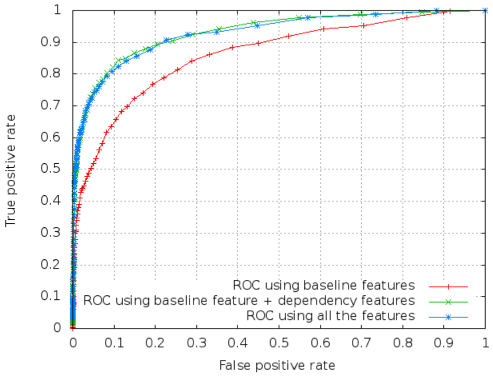

Figure 1: ROC curves for disfluency detection in true transcript

$$S_{dis}(there) = w_1 \times p_{dis}(there) + \\ w_2 \times \{p_{dis}(think\_there) + p_{dis}(there\_is)\} + \\ w_3 \times \{p_{dis}(I\_think\_there) + \\ p_{dis}(think\_there\_is) + p_{dis}(there\_is\_bad)\} \quad (1)$$

We tune the maximum span length and w_1, w_2 and w_3 on the development set. For both, the reference and the ASR transcripts, we find that the optimal span length (as tuned on our development data) is 3, and $w1$, $w2$ and $w3$ are similarly tuned to (0.6,0.1,0.3) in the reference transcripts and (0.6,0.2,0.2) in the ASR transcripts. We apply these scaling factors to posteriors estimated on test-set spans to obtain word-level decisions. The number of context words for each span is tuned to 2. Hence we use the two closest left and right neighboring words to obtain contextual features barring the fillers.

4. Results and discussion

We plot the ROC curve for word level detection of disfluencies for the baseline feature set as well as after adding each additional feature consecutively, on the reference transcripts (Figure 1) and the ASR transcripts (Figure 2). The addition of dependency features improves the ROC curves in both the cases. We observe that the results on the reference transcripts are better than the ASR transcripts in all the cases. Also, while the addition of silence features does not improve the performance in the case of reference transcripts, it improves the ROC curve for the noisy ASR transcripts. We list the area under ROC curve in Table 4 along with the detection rate at a low false alarm rate of 2% given the rare occurrence of disfluencies.

Feature set	True transcript	ASR transcript
	AUC/ Detection rate	AUC/ Detection rate
Chance	.50/2.0%	.50/2.0%
Baseline features	.86/43.4%	.72/11.0%
+ Dependency features	.93/62.1%	.82/45.2%
All features	.93/62.7%	.87/47.8%

Table 4: Results obtained on the reference and the ASR transcripts (English side of DARPA TRANSTAC data).

4.1. Discussion

The distribution of weights w_1, w_2, w_3 over the three spans suggests that tagging spans is more optimal than tagging each word individually. As the number of context words is tuned to 2, this suggests the importance of context in tagging disfluencies.

From the results we observe that the baseline features perform well above the chance model. This could be attributed to the fact that a majority of word repetitions, phrase repetitions and revisions involving part-repetitions can be well captured by the baseline features. However, the baseline features do not work as well in the case of ASR transcripts. As the LM is trained on a database comprising a majority of fluent utterances, the likelihood of a disfluency showing up in ASR transcript is low. In case of an erroneous ASR hypothesis corresponding to a disfluency, the baseline features are likely to fail.

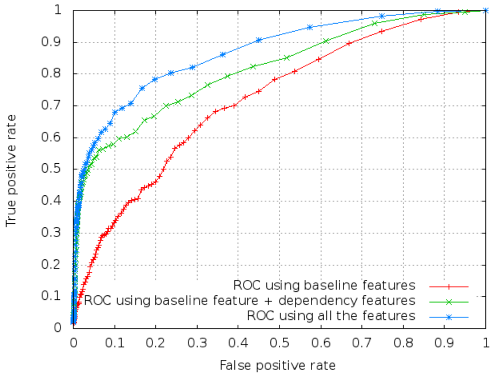

Figure 2: ROC curves for disfluency detection in ASR transcript

However, we observe that the addition of dependency based features gives a greater improvement in case of ASR transcripts over the baseline features. Unlike the baseline features, these features are expected to capture disfluencies erred as a different word during ASR transcription. Also, these features capture the syntactic flow and are more effective in capturing revisions.

When it comes to the silence feature, we observe that we hardly improve upon the baseline + dependency based features in case of reference transcripts. This indicates that in the case where oracle transcripts are available, the additional information of pausing behavior is not as useful. However, we observe a gain in case of ASR transcripts. The erroneous phrasal boundaries cause the ASR system to rely on the prosodic information to achieve better results. As we are utilizing outputs from ASR to compute silence length, we have a direct correspondence of silence lengths to the ASR hypotheses words. This helps us to train a model with a greater match towards ASR transcriptions. This variable performance of features over the two cases advocates for a more comprehensive feature design in case of ASR transcripts.

5. Conclusion

In this paper, we use a word span based technique to identify disfluencies in spoken utterances. We show that features capturing syntactic flow and pausing behavior help us improve disfluency detection over simple lexical features, notably in noisy transcripts obtained from an ASR. The ultimate goal of this work aims at identifying and removing disfluencies is in its application to spoken language technology such as spoken language translation (SLT).

As a future work, we would like to study more features that can further improve our capability to identify disfluencies specifically those introduced in the ASR transcripts. In particular, we intend to apply the disfluency detection system to an SLT system and analysis of the behavior of an ASR system around such disfluencies that can strongly boost the SLT performance. We can look at new schemes to cluster words while predicting disfluencies [20]. Finally, we can also look at prosodic nature of disfluencies to pre-process the utterances before they are fed to an ASR system [21, 22].

6. Acknowledgment

This paper is based upon work supported by the DARPA BOLT program. The views expressed here are those of the author(s) and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

7. References

[1] E. E. Shriberg, “Preliminaries to a theory of speech disfluencies,” *University of California*, 1994.

[2] D. R. Little, R. Oehmen, J. Dunn, K. Hird, and K. Kirsner, “Fluency profiling system: An automated system for analyzing the temporal properties of speech,” *Behavior research methods*, pp. 1–12, 2012.

[3] R. Eklund, “Disfluency in swedish human–human and human–machine travel booking dialogues,” *Linköping*, 2004.

[4] E.E. Shriberg, “To errrr'is human: ecology and acoustics of speech disfluencies,” *Journal of the International Phonetic Association*, vol. 31, no. 1, pp. 153–169, 2001.

[5] R. Ferreira and K. Bailey, “Disfluencies and human language comprehension,” *Trends in cognitive sciences*, vol. 8, no. 5, pp. 231–237, 2004.

[6] W. Johnson, “Measurements of oral reading and speaking rate and disfluency of adult male and female stutterers and nonstutterers.,” *The Journal of speech and hearing disorders*, p. 1, 1961.

[7] W. Wang, G. Tur, J. Zheng, and N.F. Ayan, “Automatic disfluency removal for improving spoken language translation,” in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*. IEEE, 2010, pp. 5214–5217.

[8] K. Georgila, N. Wang, and J. Gratch, “Cross-domain speech disfluency detection,” in *Proceedings of the 11th Annual Meeting of the Special Interest Group on Discourse and Dialogue*. Association for Computational Linguistics, 2010, pp. 237–240.

[9] S. Rao, I. Lane, and T. Schultz, “Improving spoken language translation by automatic disfluency removal: Evidence from conversational speech transcripts,” *Training*, vol. 6370, no. 46300, pp. 6–50, 2007.

[10] W. Wang, A. Stolcke, J. Yuan, and M. Liberman, “A cross-language study on automatic speech disfluency detection,” in *Proceedings of NAACL-HLT*, 2013, pp. 703–708.

[11] Yang Liu, Elizabeth Shriberg, Andreas Stolcke, and Mary P Harper, “Comparing HMM, maximum entropy, and conditional random fields for disfluency detection.,” in *INTERSPEECH*. Citeseer, 2005, pp. 3313–3316.

[12] V. Rangarajan and S. Narayanan, “Analysis of disfluent repetitions in spontaneous speech recognition,” *Proc. EUSIPCO 2006*, 2006.

[13] Matthew Lease, Mark Johnson, and Eugene Charniak, “Recognizing disfluencies in conversational speech,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 5, pp. 1566–1573, 2006.

[14] M. Honal and T. Schultz, “Automatic disfluency removal on recognized spontaneous speech-rapid adaptation to speaker dependent disfluencies,” in *Proc. of ICASSP*, 2005, vol. 1, pp. 969–972.

[15] D. Stallard, R. Prasad, P. Natarajan, F. Choi, S. Saleem, R. Meermeier, K. Krstovski, S. Ananthakrishnan, and J. Devlin, “The BBN transtalk speech-to-speech translation system,” *Speech and Language Technologies, InTech*, pp. 31–52, 2011.

[16] Long Nguyen and Richard Schwartz, “Efficient 2-pass n-best decoder,” in *Fifth European Conference on Speech Communication and Technology*, 1997.

[17] W. Chen, S. Ananthakrishnan, R. Prasad, and Natarajan P., “Variable-span out-of-vocabulary named entity detection,” in *Fourteenth Annual Conference of the International Speech Communication Association*. IEEE, 2013.

[18] J. Nocedal, “Updating quasi-newton matrices with limited storage,” *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.

[19] Stanford Parser Version, “1.6 (2008),” *Chicago Ill.: SPSS Inc.*

[20] W. Schuler, S. AbdelRahman, T. Miller, and L. Schwartz, “Broad-coverage parsing using human-like memory constraints,” *Computational Linguistics*, vol. 36, no. 1, pp. 1–30, 2010.

[21] M. Kaushik, M. Trinkle, and A. Hashemi-Sakhtsari, “Automatic detection and removal of disfluencies from spontaneous speech,” in *Proc. 13th Australasian Int. Conf. on Speech Science and Technology Melbourne*, 2010, pp. 98–101.

[22] Y. Liu, E. Shriberg, A. Stolcke, D. Hillard, M. Ostendorf, and M. Harper, “Enriching speech recognition with automatic detection of sentence boundaries and disfluencies,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 5, pp. 1526–1540, 2006.