Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)

Automatic Emphatic Information Extraction from Aligned Acoustic Data and Its Application on Sentence Compression

Yanju Chen
Department of Computer Science
Sun Yat-sen University
Guangzhou, China
chyanju@gmail.com

Rong Pan*
Department of Computer Science
Sun Yat-sen University
Guangzhou, China
panr@sysu.edu.cn

Abstract

We introduce a novel method to extract and utilize the semantic information from acoustic data. By automatic Speech-To-Text alignment techniques, we are able to detect word-based acoustic durations that can prosodically emphasize specific words in an utterance. We model and analyze the sentence-based emphatic patterns by predicting the emphatic levels using only the lexical features, and demonstrate the potential ability of emphatic information produced by such an unsupervised method to improve the performance of NLP tasks, such as sentence compression, by providing weak supervision on multi-task learning based on LSTMs.

Introduction

Specific words can be prosodically emphasized in an utterance by a speaker in order to draw attentions on them, which can be modeled by pitch accents of words (Bolinger 1958). Also referred as prosodic prominence, pitch accent is found to emphasize several semantic information in an utterance such as uncertainty, contrast, turn-taking cues and so on, whose changes in an utterance can be perceived by listeners and thus convey certain kinds of emphasis (Terken 1991). The detection of prosodic prominence shows improvements on different tasks, such as Text-to-Speech synthesis and spoken language summarization. With most of the detections of prosodic prominence are done by using acoustic features (acoustic durations and intensities, extremity of fundamental frequency minima and maxima), there are also works investigating predictions of emphatic words using only lexical features (Brenier, Cer, and Jurafsky 2005; Brenier 2008), which shows promising results and potential improvements on more NLP tasks but are partly restricted by the cost of high-quality manual feature extraction.

As one of the standard NLP tasks, the target of sentence compression is to generate shorter paraphrases of sentences, which can be further used both to assist other tasks such as automatic summarization (Berg-Kirkpatrick, Gillick, and Klein 2011) and to provide assistive applications for poor readers (Canning et al. 2000), as well as to generate readable news headlines (Filippova 2010). In sentence compression systems that deal with deletion-based compression, generated words constitute subsequences of the input sequences. Existing systems mostly rely heavily on syntactic information (McDonald 2006; Berg-Kirkpatrick, Gillick, and Klein 2011), resulting in a vulnerability to error propagation. Thus, competitive systems without any syntactic information are proposed (Filippova et al. 2015), which benefits from LSTM structures and efficient heuristic search in the scope of neural network sequence labeling, providing possible solutions to the weakness of the traditional methods. More recent advances include using eye-tracking recordings (Barrett, Agić, and Søgaard 2015) to improve performance of LSTM-based sentence compression models (Klerke, Goldberg, and Søgaard 2016).

In this paper we address the following question: can useful emphatic information be automatically extracted from the prevailing acoustic data without any manual feature extraction and be used to help improve the performance of natural language processing tasks such as sentence compression? While sentence compression requires the models of a good comprehension of the semantic context and the exact intention of the input sentence, we believe the supervision of additional emphatic data can be a boost to the later, and the LSTM structures dealing with the former, which will be supported by our evidence. Meanwhile, with the Speech-To-Text alignment techniques, we present a faster approach to automatically extract approximate emphatic patterns from aligned acoustic data, thus lowering the cost of manual feature extraction in emphatic words detection and prediction and providing weak supervision as an auxiliary task to improve sentence compression performance using LSTM structures.

The contributions of our work are summarized as follows:

- We propose a faster approach to automatically extract the emphatic information from prevailing aligned acoustic data.
- We model and analyse the extracted emphatic patterns and demonstrate how the disjoint emphatic data can be added to improve the performances of sentence compression tasks using LSTM structures.
- We observe competitive improvements on our multi-task sentence compression models with the disjoint emphatic data, compared with the baselines.

*Corresponding Author

Related Works

There are some related works on sentence compression and the detection and prediction of emphatic words respectively. However, to our knowledge, there is no competitive research that connects the two tasks together and improve the former with a weak supervision from the later. There are several works that prove the benefits of deep multi-task learning with low level tasks supervised at lower layers (Søgaard and Goldberg 2016), and works that improve sentence compression by introducing gaze measures prediction as an auxiliary task (Klerke, Goldberg, and Søgaard 2016), and therefore we argue that emphatic patterns of sentences from acoustic data can also bring improvements even when they are extracted automatically without manual annotation.

Emphatic Words Prediction from Text

While several classifiers for prominence detection and prediction have been proposed (Chen, Hasegawa-Johnson, and Cohen 2004; Sun 2002; Brenier, Cer, and Jurafsky 2005; Hovy et al. 2013), with some of them utilizing lexical features, most of the features used require labeling by experienced human annotators, and specifically for lexical features, the manual alignment of text and speech are required. The labeling cost is thus high especially for some of the advanced linguistic text features that only professional annotators can afford (Brenier 2008). As a result, the overall accuracies of these classifiers are high and this provides better improvements in further applications like text-to-speech synthesis.

Specifically for those classifiers that predict prominence from only lexical features, the overall accuracies can be around 79.7% (Brenier, Cer, and Jurafsky 2005) using maximum entropy classifiers. Since the weak supervision method is considered a new possibility to solve classification problems with different constraints in the access to the class information (Hernández-González, Inza, and Lozano 2016), a balance between the accuracy and the quantity of the data can then be reached to provide sufficient information for many NLP tasks (Hoffmann et al. 2011), which indicates that an approximate approach that generates a larger but less accurate corpus and consumes less human labors can be considered in the perspective of natural language processing.

Robust Sentence Compression without Linguistic Information

Recent advances of sentence compression involve the removal of syntactic information (no Part-of-Speech tags, Name Entity tags or dependencies). A competitive system (Filippova et al. 2015) is proposed using only word embeddings and gold-standard labels of previous words during training (generated labels during decoding). The model outperforms the baseline (McDonald 2006), which employs syntactic information. Stacked LSTMs together with beam search on top layer generate competitive results. Another model (Klerke, Goldberg, and Søgaard 2016) uses stacked Bi-LSTMs with eye-tracking measures as auxiliary task is also reported competitive results, but the model has CCG-tags prediction as a second auxiliary task and is not strictly a model without syntactic information.

Our model for experiment is similar to the two above, with no additional heuristic search or syntactic information attached, which is a pure multi-task learning neural network.

Emphatic Words Detection in Acoustic Data

Usually acoustic data is a good resource for emphatic words detection. In this paper, we refer ***emphatic words*** to those words that are *a subset of pitch accents that have been shown to be categorically interpreted as distinct from neural pitch accents* (Ladd and Morton 1997) and *conveying an acute degree of emphasis* (Brenier, Cer, and Jurafsky 2005).

Semantic Information in Prosodic Prominence

Prosodic prominence is found to convey various semantic information (Pon-Barry and Shieber 2009; Wang et al. 2013; Cao et al. 2014; Ferguson, Durrett, and Klein 2015; Levitan et al. 2016) through changes between words and syllables. Such information including contrast, incredulity, uncertainty, adverbial focus and so on, correlates with the speaker's intentions and can be perceived by listeners. Major indicators to judge whether a word is pitch accented or not include duration, intensity and extremity of fundamental frequency minima and maxima. The detection of the three indicators are usually done by annotators using several acoustic tools according to general standards before the annotated data can be used to train an automatic prominence classifier.

Automatic Speech-To-Text Alignment

Also known as forced alignment, automatic speech-to-text alignment has received some attention for different research goals. Given an exact transcription of what is being spoken in the acoustic data, the aligner is asked to identify the time when each word in the transcription was spoken in the utterance.

Among the three indicators of prosodic prominence, word duration is a relatively easier indicator to be detected using an automatic speech-to-text alignment system, because an aligner provides the exact time that a transcribed word occurs and thus we can estimate the word's duration from the aligner's outputs, thus making it possible to collect emphatic data in a faster and unsupervised way, as well as modeling the emphatic patterns of a given aligned sentence. An automatic data collection procedure can then be designed to establish a large scale corpus of emphatic data. We will examine later how the information extracted based on a single indicator can help improve certain NLP tasks.

Gathering Emphatic Patterns from Aligned Acoustic Data

We collect acoustic data with corresponding transcriptions from *Lit2Go*[1], a free online collection of stories and poems in audiobook format with transcriptions. There are over 200 books read by distinct readers. Each book is segmented to several passages, and we collect over 4,000 passages and

[1] http://etc.usf.edu/lit2go/

align them in *cmusphinx*[2] to establish the emphatic corpus. Table 1 shows more details about the data.

Table 1: Basic Information of Collected Acoustic Data

authors:	208	books:	205
genres:	22	passages:	4198
sentences:	286,083	words:	5,881,720
vocab. size:	48,204	mean sent. len.:	20

The General Process of Aligning Data

First, we do a pre-processing on the transcriptions, including a separation of the punctuations from the attached words and so on. Second, we feed the aligner with acoustic data and their corresponding transcriptions. Third, we calculate the ***standard duration (S.Duration)*** of each word according to Eq. (1). Finally, we extract each sentence with the corresponding *S.Duration* sequence from the aligned data as an ***emphatic pattern***.

$$S.Duration = \frac{total\ time\ duration}{num.\ of\ syllables}. \quad (1)$$

We define an ***emphatic pattern*** here as an ordered sequence of standard duration of a sentence. To give a faster estimation of each word's acoustic duration, we adopt the assumption that each syllable has an equal length of time duration in Eq. (1), so that we can alleviate the effects of word length on the word's total time duration by considering the number of syllables a word has.

Thus, a typical entry in the corpus of emphatic data is a pair of two components: a lexical sentence and a corresponding emphatic pattern. There are over 280,000 pairs in the corpus.

Modeling Emphatic Patterns in Aligned Text

We extend the view of emphatic words prediction from modeling the context of a single word to modeling the whole sentence's emphatic pattern. The Long Short Term Memory (Hochreiter and Schmidhuber 1997) structure can model larger context as well as controlling the access of error signals, making the learning of sentence-level information and long-term dependencies possible. We use only lexical features of the emphatic data to predict the emphatic patterns in a sequence labeling manner.

Generating Word-Based Emphatic Levels

We normalize and binarize the words' standard durations in a sentence to alleviate the effects of different reading styles (e.g., speeds, tones) of readers. The standard durations within a sentence will first be normalized and mapped to a list of integer ***emphatic levels***, which denotes the relative standard duration of each word in a sentence ranging from 1 to 30. According to the changes between the emphatic levels, they are then binarized to be 0 (not emphatic) or 1 (emphatic), which we denote as ***binary emphatic levels***.

[2]http://cmusphinx.sourceforge.net/

Algorithm 1 Binary Emphatic Levels Generation

```
// BinEmpSeqs: binary emphatic levels of sentences
// EmpLvs: emphatic levels of one sentence
// BinEmpLvs: binary emphatic levels of one sentence
// elv_{-1}: previous word's emphatic level
// bslot_{-1}: previous word's binary emphatic level
BinEmpSeqs = []
for emphatic pattern of each sequence do
  // 1. generate emphatic levels of a single sentence
  P_5  = 5th percentile in the emphatic pattern
  P_95 = 95th percentile in the emphatic pattern
  EmpLvs = []
  for each standard duration sd do
    if sd > P_95 then
      cslot = 30
    else if sd < P_5 then
      cslot = 1
    else
      cslot = round(2 + 28 * (sd - P_5)/(P_95 - P_5))
    end if
    EmpLvs.add(cslot)
  end for
  // 2. binarize emphatic levels of a single sentence
  BinEmpLvs = []
  for each emphatic level elv in EmpLvs do
    if elv - elv_{-1} > 4 && elv > 15 then
      bslot = 1
    else if elv_{-1} - elv < 2 && elv > 15
          && bslot_{-1} == 1 then
      bslot = 1
    else
      bslot = 0
    end if
    BinEmpLvs.add(bslot)
  end for
  BinEmpSeqs.add(BinEmpLvs)
end for
```

Alg. 1 shows the details of how a sequence of binary emphatic levels is generated in every sentence. As a post processing, binary emphatic levels of those words on a stopword list or in the first positions of sentences will be set to 0 before the data is used for the emphatic words prediction model. Several labeling results are shown in Table 2 after the post processing is done.

Emphatic Words Prediction with LSTMs

LSTM is used to model the binary emphatic patterns. Let $\boldsymbol{X}$ be the input sequence and $\boldsymbol{A}$ be the output emphatic patterns, as defined below:

$$\boldsymbol{X} = (x_1, ..., x_N), \boldsymbol{A} = (a_1, ..., a_N).$$

We are to optimize the following problem:

$$\theta^* = \arg\max_{\theta} \sum_{X,A} \log p(A|X;\theta). \quad (2)$$

For the basic LSTM model depicted in Figure. 1, p can be

Table 2: Example Binary Emphatic Levels Generated from Acoustic Data after Post Processing

Type	Sample Labeling Results
V	Jim was **laid up** for **four days** and **nights** .
L	18 12 20 30 17 22 26 11 21 .
V	But it was **too dark** to **see** yet , **so** we **made** the **canoe fast** and **set in** her **to** wait for **daylight** .
L	12 9 11 17 20 11 30 17 23 11 17 8 16 25 14 19 18 13 22 17 5 22
V	I didn't need anybody to **tell me** that **that** was an **awful bad sign** and would **fetch me** some **bad luck** , so I was
L	30 9 14 14 13 18 28 13 22 14 11 18 25 30 13 9 27 28 14 19 24 , 20 22 10
V	**scared** and most **shook** the **clothes off** of **me** .
L	25 7 12 24 9 16 20 7 30 .

[1] "V" means visualized results, and "L" means original emphatic levels.
[2] Underlined bold-faced words are emphatic, otherwise non-emphatic.

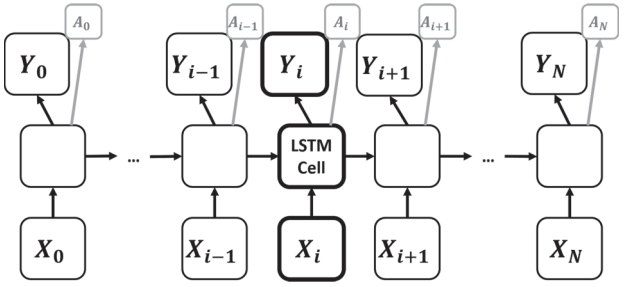

Figure 1: Basic LSTM Unrolled Through Time

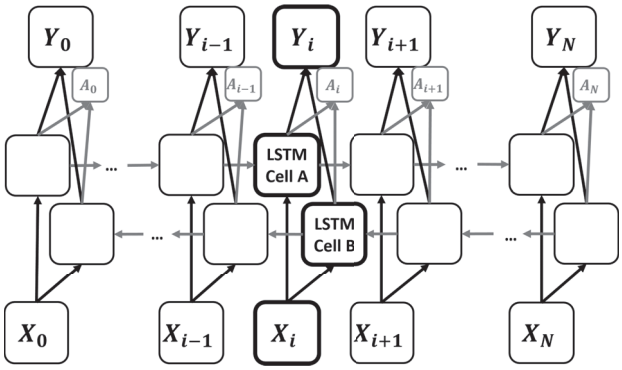

Figure 2: Bi-Directional LSTM Unrolled Through Time

decomposed as follows:

$$p(A|X;\theta) = \prod_{t=1}^{N} p(A_t|X_0, X_1, ..., X_t; \theta). \quad (3)$$

For the bi-directional LSTM model depicted in Fig. 2, p can be decomposed as follows:

$$p(A|X;\theta) = \prod_{t=1}^{N} p_f(A|X;\theta) p_b(A|X;\theta), \quad (4)$$

where

$$p_f(A|X;\theta) = \prod_{t=1}^{N} p_f(A_t|X_0, X_1, ..., X_t; \theta), \quad (5)$$

$$p_b(A|X;\theta) = \prod_{t=1}^{N} p_b(A_t|X_N, X_{N-1}, ..., X_t; \theta). \quad (6)$$

Using the optimal θ^*, the prediction can then be estimated:

$$\hat{A} = \arg\max_{A} p(A|X;\theta^*). \quad (7)$$

We use LSTM in the experiment of emphatic words prediction (see Fig. 1), with a shared softmax classifier connected to each hidden state that predicts the binary emphatic labels of the current time step. We also use bi-directional LSTM (see Fig. 2) as a further observation. Adadelta (Zeiler 2012) is used to maximize the training objective.

We used pre-trained word embeddings from the skip-gram model[3] (Mikolov and Dean 2013) for every input word. Specifically, to enhance the robustness of the model, the following operations are applied:

- Length of input sequences will be at most 50 words, otherwise cut short.
- The embeddings only include words that appear no less than 2 times in the data.
- We use <UKN> to represent words that does not appear in the embeddings.

Training, validating and testing sets are split. Table 3 shows the information of the experiment settings.

Results

We measure and record the following two metrics:

- Word-Based Accuracy ($W.Acc$): how many words are correctly labeled
- Sentence-Based Accuracy ($S.Acc$): how many sentence are fully correctly labeled

We record a word-based accuracy of 82.90% and a sentence-based accuracy of 9.47% on the test set, within 10 training epochs. The basic LSTM sequence labeling model shows a potential capability of capturing and predicting the emphatic patterns. The Bi-LSTM model has recorded 85.24% word-based accuracy and 14.42% sentence-based accuracy.

[3] https://code.google.com/p/word2vec/

Improvements on Sentence Compression Task

To further verify the potential capability of the extracted semantic information over specific NLP tasks, we carry out a series of experiments on deletion-based sentence compression tasks to evaluate the performance of several basic models when emphatic data is added as an auxiliary task.

Sentence Compression Using Multi-Task LSTMs

Similar to the models used by Klerke et al. (2016), we adopt even simpler model structures in order to better investigate the actual effects of the emphatic data over the whole task, without either the prediction of CCG tags or the CASCADED-LSTM structure. We follow part of the task settings (Filippova et al., 2015) which introduce no additional syntactic or linguistic information into the models but only pre-trained embeddings processed with similar operations in the emphatic words prediction experiment.

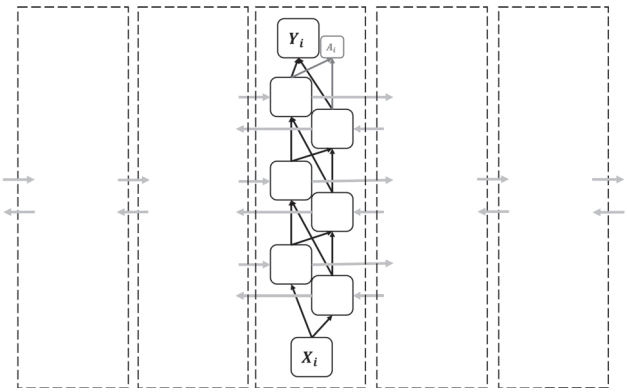

Figure 3: Unrolled Stacked Bi-LSTM High-Level Overview

A bi-directional LSTM reads a sequence in both regular and reversed orders. Our basic model structure is depicted in Figs. 1 and 2. For some larger datasets, we use 3-layered stacked bi-directional LSTM to capture deep semantic information from the sequences, as shown in Fig. 3. Two different softmax classifiers are connected to each of the hidden states (connected to the top hidden states for stacked bi-directional LSTM) to perform multi-task learning. One for the classification of task data and the other for the emphatic data (or other extra data for comparison).

Additionally for sentence compression, we define $\boldsymbol{Y}$ as the output compression labels (1 if a word is retained, 0 if deleted) as follows:

$$\boldsymbol{Y} = (y_1, ..., y_N).$$

Besides the auxiliary optimization problem defined in Eq. (2), our major optimization problem here is:

$$\theta^* = \arg\max_{\theta} \sum_{X,Y} \log p(Y|X;\theta). \quad (8)$$

The sentence compression predictions can be estimated by:

$$\hat{Y} = \arg\max_{Y} p(Y|X;\theta^*). \quad (9)$$

The parameters θ are shared by the two tasks.

Experiments

We carry out comparative experiments. The performance of a baseline model is compared with a multi-task version that has emphatic data as an auxiliary task. Additionally we carry out another set of experiments with the same multi-task model but different extra data, as a further comparison.

Table 3: Datasets Characteristics

Dataset	Train	Valid	Test	Del.Rate[1]
GOOGLE	8,000	1,000	1,000	0.59
BROADCAST	880	78	412	0.33
Emphatic[2]	230k*	25k*	28k*	0.73

* These are approximate numbers.
[1] Sentences with more than 50 words are cut short.
[2] For sentence compression, the whole set is used as an auxiliary task; for emphatic words prediction, the whole data set is then split as above.

Compression Data We use two different sentence compression datasets, details shown in Table 3:

- The publicly available subset of GOOGLE[4]
- BROADCAST (Clarke and Lapata 2006)

Baselines As shown in Figs. 1 and 2, we use LSTM and bi-directional LSTM for the compression data, and additionally we evaluate the 3-layered stacked bi-directional LSTM on the GOOGLE (Filippova et al. 2015) dataset. There are no auxiliary tasks in the three baseline models.

Auxiliary Tasks We evaluate the emphatic data as an auxiliary task in the multi-task learning models. Meanwhile, the first pass duration of eye-tracking data (denoted as Gaze.fp in tables) is also used as an auxiliary task in another comparative experiment, pre-processed by Klerke (2016).

Evaluation Metrics For different deletion rates (after the pre-processing) of datasets shown in Table 3, the F1-Scores of both label 0 and 1 are of equal importance and thus we evaluate both F1-Score of label 0 ($F1_0$) and F1-Score of label 1 ($F1_1$). We also evaluate the *word-based accuracy* ($W.Acc$) and the *sentence-based accuracy* ($S.Acc$).

Multi-Task Learning In each training epoch, the model is first fed with certain amount of random samples from data of the auxiliary task, and then certain amount of samples from the compression task. The process may be repeated once in the same epoch (see Table 4 for more details).

Other Settings The dimensions of input and hidden layers and the embeddings (pre-trained and pre-processed, see previous section) are 300, and at the output layer we predict sequences of two categories (compression task and emphatic task) or six categories (eye-tracking task). All models are trained for 30 iterations and we adopt an early-stopping strategy to select parameters with highest word-based accuracy ($W.Acc$) on validation set and perform and record evaluations on testing set. We run each experiment for 32 times. The average of each evaluation metrics is recorded and we also perform significance tests between the sample metrics

[4] http://storage.googleapis.com/sentencecomp/compression-data.json

Table 4: Feeding Order in An Epoch

Experiment	LSTM	Bi-LSTM	Stacked Bi-LSTM
Baseline[1]	C:Full	C:Full	C:Full
Emphatic &Gaze.fp[2]	A:2500 C:1500 A:2500 C:1500	A:2000 C:2000 A:2000 C:2000	A:2000 C:2000 A:2000 C:2000
Emphatic &Gaze.fp[3]	A:2500 C:Full	A:2000 C:Full	- -

* "Full" means all 8,000 training samples in GOOGLE dataset, all 880 training samples in BROADCAST dataset, otherwise randomly sampled; "C" means samples from compression task; "A" means samples from auxiliary task.
* Samples are fed in the order as shown in each cell from top to down alternately.
[1] The rules apply to all datasets.
[2] The rules apply to GOOGLE dataset.
[3] The rules apply to BROADCAST datasets.

drawn from the baseline system and from the comparative systems to give confidence levels.

Results and Discussion

Our results are presented below. Across all datasets, the emphatic data leads to improvements over the baselines in all evaluation metrics. As the structures become deeper and more complex, when capturing more contexts, the improvements are much more significant.

- GOOGLE Dataset

Table 5: Performance on GOOGLE Dataset

Model	Data	GOOGLE			
		$W.Acc$	$F1_0$	$F1_1$	$S.Acc$
LSTM	Baseline	79.15	82.73	73.70	6.51
	Emphatic	**79.40**	**82.86**	**74.19**	**7.18**
	Gaze.fp	79.27	82.84	73.81	6.58
Bi-LSTM	Baseline	79.79	83.31	74.40	7.19
	Emphatic	**80.14**	**83.73**	**74.50**	**8.11**
	Gaze.fp	79.71	83.40	73.97	7.53
Stacked Bi-LSTM	Baseline	79.94	83.40	74.63	8.00
	Emphatic	**80.30**	**83.74**	**74.99**	**9.26**
	Gaze.fp	79.95	83.48	74.50	8.53

As shown in Table 5, all models with emphatic data as an auxiliary task outperform their comparative models in all evaluation metrics with significant performances. We observed that in all three LSTM models, as an important indicator, the sentence-based accuracy ($S.Acc$) has more than 10% improvements over the baselines. As a result, the emphatic data coordinates with the GOOGLE dataset's aggressive compressions, imposing a positive regularization during multi-task training.

- BROADCAST Datasets

The BROADCAST datasets are manually annotated by three different annotators. We observe significant improvements on nearly every metric on bi-directional LSTM models, while the LSTM models don't show much significant improvements, as shown in Tables 6 , 7 and 8. Specifically, we observe that the emphatic data always provides a solid regularization over the whole sentences, resulting in steady improvements on sentence-based accuracies.

Table 6: Performance on BROADCAST1 Dataset

Model	Data	BROADCAST1			
		$W.Acc$	$F1_0$	$F1_1$	$S.Acc$
LSTM	Baseline	72.28	14.56	83.43	**10.93**
	Emphatic	**72.70**	**19.37**	83.53	10.87
	Gaze.fp	72.69	18.56	**83.56**	10.90
Bi-LSTM	Baseline	72.76	21.17	83.51	11.30
	Emphatic	**73.56**	**25.93**	**83.87**	**11.95**
	Gaze.fp	73.34	23.98	83.82	11.81

Table 7: Performance on BROADCAST2 Dataset

Model	Data	BROADCAST2			
		$W.Acc$	$F1_0$	$F1_1$	$S.Acc$
LSTM	Baseline	79.10	13.27	88.12	22.19
	Emphatic	79.34	**17.20**	88.19	**22.25**
	Gaze.fp	**79.42**	15.98	**88.27**	22.12
Bi-LSTM	Baseline	79.78	22.89	88.35	22.82
	Emphatic	**80.37**	**26.60**	**88.66**	**23.19**
	Gaze.fp	80.24	26.11	88.59	22.97

Table 8: Performance on BROADCAST3 Dataset

Model	Data	BROADCAST3			
		$W.Acc$	$F1_0$	$F1_1$	$S.Acc$
LSTM	Baseline	66.85	36.22	**77.55**	9.60
	Emphatic	67.06	37.93	77.52	**9.70**
	Gaze.fp	**67.19**	**40.38**	77.34	8.56
Bi-LSTM	Baseline	67.58	**38.94**	77.86	11.48
	Emphatic	**68.35**	38.39	**78.66**	**11.65**
	Gaze.fp	68.23	38.01	78.59	11.57

Conclusion and Future Work

We present, to our knowledge, a first attempt at the automatic extraction of the semantic information from acoustic data to help improve sentence compression task. The results of our experiments indicate the potential ability of the aligned acoustic data when modeled to capture emphatic information in the text and embedded as an auxiliary task during multi-task learning. The faster approach to extract emphatic patterns proves to work well and generate improvements in related experiments. The weak supervision of the emphatic data provides positive regularization to many of the training process.

There remain many improvements in our work. For example, the automatic extraction can be better tuned according to different tasks, and a more sophisticated multi-task training manner can be thus applied to make better use of the regularization effects of the emphatic data.

Acknowledgments

We would like to thank the many referees of the previous version of this paper for their extremely useful suggestions and comments. Thanks to our colleagues for helpful discussions, and to anonymous reviewers for their suggestions for improving the paper.

References

Barrett, M.; Agić, Ž.; and Søgaard, A. 2015. The dundee treebank. In *The 14th International Workshop on Treebanks and Linguistic Theories (TLT 14)*.

Berg-Kirkpatrick, T.; Gillick, D.; and Klein, D. 2011. Jointly learning to extract and compress. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 481–490. Association for Computational Linguistics.

Bolinger, D. L. 1958. A theory of pitch accent in english. *Word* 14(2-3):109–149.

Brenier, J. M.; Cer, D. M.; and Jurafsky, D. 2005. The detection of emphatic words using acoustic and lexical features. In *INTERSPEECH*, 3297–3300.

Brenier, J. M. 2008. *The Automatic Prediction of Prosodic Prominence from Text*. ProQuest.

Canning, Y.; Tait, J.; Archibald, J.; and Crawley, R. 2000. Cohesive generation of syntactically simplified newspaper text. In *International Workshop on Text, Speech and Dialogue*, 145–150. Springer.

Cao, H.; Benus, S.; Gur, R. C.; Verma, R.; and Nenkova, A. 2014. Prosodic cues for emotion: analysis with discrete characterization of intonation. *Speech prosody 2014*.

Chen, K.; Hasegawa-Johnson, M.; and Cohen, A. 2004. An automatic prosody labeling system using ann-based syntactic-prosodic model and gmm-based acoustic-prosodic model. In *Acoustics, Speech, and Signal Processing, 2004. Proceedings.(ICASSP'04). IEEE International Conference on*, volume 1, I–509. IEEE.

Clarke, J., and Lapata, M. 2006. Constraint-based sentence compression an integer programming approach. In *Proceedings of the COLING/ACL on Main conference poster sessions*, 144–151. Association for Computational Linguistics.

Ferguson, J.; Durrett, G.; and Klein, D. 2015. Disfluency detection with a semi-markov model and prosodic features. In *Proc. NAACL HLT*.

Filippova, K.; Alfonseca, E.; Colmenares, C.; Kaiser, L.; and Vinyals, O. 2015. Sentence compression by deletion with lstms.

Filippova, K. 2010. Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, 322–330. Association for Computational Linguistics.

Hernández-González, J.; Inza, I.; and Lozano, J. A. 2016. Weak supervision and other non-standard classification problems: a taxonomy. *Pattern Recognition Letters* 69:49–55.

Hochreiter, S., and Schmidhuber, J. 1997. Long short-term memory. *Neural computation* 9(8):1735–1780.

Hoffmann, R.; Zhang, C.; Ling, X.; Zettlemoyer, L.; and Weld, D. S. 2011. Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, 541–550. Association for Computational Linguistics.

Hovy, D.; Anumanchipalli, G. K.; Parlikar, A.; Vaughn, C.; Lammert, A. C.; Hovy, E. H.; and Black, A. W. 2013. Analysis and modeling of "focus" in context. In *INTERSPEECH*.

Klerke, S.; Goldberg, Y.; and Søgaard, A. 2016. Improving sentence compression by learning to predict gaze. *arXiv preprint arXiv:1604.03357*.

Ladd, D. R., and Morton, R. 1997. The perception of intonational emphasis: continuous or categorical? *Journal of Phonetics* 25(3):313–342.

Levitan, S. I.; An, G.; Ma, M.; Levitan, R.; Rosenberg, A.; and Hirschberg, J. 2016. Combining acoustic-prosodic, lexical, and phonotactic features for automatic deception detection. *Interspeech 2016* 2006–2010.

McDonald, R. T. 2006. Discriminative sentence compression with soft syntactic evidence. In *EACL*.

Mikolov, T., and Dean, J. 2013. Distributed representations of words and phrases and their compositionality. *Advances in neural information processing systems*.

Pon-Barry, H., and Shieber, S. 2009. The importance of sub-utterance prosody in predicting level of certainty. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, 105–108. Association for Computational Linguistics.

Søgaard, A., and Goldberg, Y. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *The 54th Annual Meeting of the Association for Computational Linguistics*, 231.

Sun, X. 2002. Pitch accent prediction using ensemble machine learning. In *INTERSPEECH*.

Terken, J. 1991. Fundamental frequency and perceived prominence of accented syllables. *The Journal of the Acoustical Society of America* 89(4):1768–1776.

Wang, W. Y.; Biadsy, F.; Rosenberg, A.; and Hirschberg, J. 2013. Automatic detection of speaker state: Lexical, prosodic, and phonetic approaches to level-of-interest and intoxication classification. *Computer Speech & Language* 27(1):168–189.

Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.