USING SPOKEN UTTERANCE COMPRESSION FOR MEETING SUMMARIZATION: A PILOT STUDY

Fei Liu and Yang Liu

Computer Science Department
The University of Texas at Dallas
{feiliu, yangl}@hlt.utdallas.edu

ABSTRACT

Most previous work on meeting summarization focused on extractive approaches; however, directly concatenating the extracted spoken utterances may not form a good summary. In this paper, we investigate if it is feasible to compress the transcribed spoken utterances and if using the compressed utterances benefits meeting summarization. We model the utterance compression task as a sequence labeling problem, and show satisfying performance using a CRF model that incorporates a variety of features capturing lexical, syntactic, and discourse information. We evaluate the impact of utterance compression on the meeting summarization task using compressed sentences (pre-compression) and original transcripts (post-compression), and find that using the compressed meeting transcripts yields slightly better summarization performance. In general, using sentence compression together with extractive summarization can generate reasonable compressed summaries. This is a step closer to abstractive summarization.

Index Terms— Meeting Summarization, Utterance Compression, Conditional Random Fields, MMR, ICSI Meeting Corpus

1. INTRODUCTION

Most of the current summarization systems adopt extractive approaches. The system first extracts a set of salient sentences that can best convey the main content of the text document or the spoken audio file; these sentences are then concatenated into a summary according to their appearance in the original document. For the well-formed written text domain, this approach results in quite good summary quality, since the extracted sentences themselves are usually well-formed, self-explainable, and have good sentence and discourse structure. On the other hand, directly concatenating the transcribed utterances may not form a good summary for speech domains. This is especially true for meetings, where disfluencies and redundancies in spontaneous speech significantly affect the readability of the extracted summary. In this paper, we propose to pipe a spoken utterance compression module with an extractive meeting summarization system, with the expectation of generating condensed meeting summaries and improving summary quality.

In Table 1, we show an example of the extractive summary and its compressed variant for a meeting dialogue segment. The "Original Extractive Summary" was formed by directly concatenating the extracted summary sentences (using human transcripts). The "Compressed Summary" was generated by manually compressing the extractive summary at the sentence level. We can see that the quality of the original extractive summary is not very good. In contrast, the compressed summary removes many unnecessary words from the original extractive summary. It effectively highlights the main content and its readability is much better. In this sense, the compressed meeting summary is also closer to abstractive meeting summaries. For abstractive summarization, we may apply sentence compression techniques to extracted summary sentences, followed by further sentence merging, compaction, and generation.

Original Extractive Summary	
423	there there are a variety of ways of doing it
433	so it's possible that we could do something like a summary node of some sort that
444	so what i was gonna say is is maybe a good at this point is to try to informally
446	i mean not necessarily in th- in this meeting but to try to informally think about what the decision variables are
450	and the other trick which is not a technical trick it's kind of a knowledge engineering trick is to make the n- -pau- each node sufficiently narrow that you don't get this combinatorics
Compressed Summary	
423	there are ways of doing it
433	it's possible we could do a summary node
444	good at this point is to try informally
446	to informally think about what the decision variables are
450	make each node sufficiently narrow that you don't get this combinatorics

Table 1. Human compressed summary sentences for an example meeting dialogue segment. Dialogue act indices (based on the entire meeting) are shown in the first column.

In this paper, we investigate automatically generating compressed meeting summaries by piping the spoken utterance compression module with a maximum marginal relevance (MMR) based extractive meeting summarization system. Two key questions arise in this process. First, is it possible to automatically compress spoken utterances with reasonable performance? To investigate this question, we formulate the utterance compression task as a sequence labeling problem. We first collect a large set of manually compressed spoken utterances, then utilize the conditional random fields (CRF) model to automatically determine whether a word should be kept in the compressed utterance or not. The second question is, should we use the original or the compressed sentences for summary sentence selection? Under the extractive summarization framework, we compare pre-compression and post-compression settings, and eval-

uate the system performance against the human selected original summary sentences as well as human compressed summaries.

The contributions of this study are: (1) we form a corpus of human compressed spoken utterances, which is comparable in size with other similar corpora on written text domain (Section 3); (2) we propose to use a CRF model for automatic sentence compression that incorporates word identity features, part-of-speech (POS) features, position features, and features derived from both syntactic and discourse parsing trees (Section 4); (3) we evaluate the impact of using compressed sentences on summarization, and propose a fully automatic summarizer that generates compressed meeting summaries by combining the utterance compression module with an extractive summarization system (Section 5).

2. RELATED WORK

Many extractive summarization approaches have been evaluated for speech summarization. [1] applied the MMR approach to extract salient sentences from dialogue segments and achieved satisfying performance. [2, 3] proposed a concept-based integer linear programming (ILP) framework for meeting summarization and demonstrated competitive results. With respect to the supervised approaches, [4, 5, 6, 7, 8] incorporated lexical, structural, and prosodic information (such as pitch, duration, energy, and pause) in the supervised framework for speech summarization. [9] proposed a risk minimization framework for extractive speech summarization.

There is also some work on speech summarization that focused on generating condensed representation of summaries. [10] proposed to extract a set of words from automatically transcribed speech that maximizes a summarization score consisting of word significance measure, confidence score, linguistic likelihood, and a word concatenation probability. [11] proposed to generate abstracts of meeting conversations based on general conversation ontology. They also showed that users prefer abstract-style summaries over extracts. [12] experimented with ILP and lexicalized Markov grammar based sentence compression approaches to compress the human annotated meeting summaries. Different from the above work, in this paper we develop a fully automatic summarizer to generate compressed meeting summaries. We investigate using a CRF model for utterance compression that incorporates a rich set of features extracted from words, POS, position, syntactic parses, and discourse dependencies.

3. DATA AND ANNOTATION

We used 26 meetings from the ICSI corpus [13, 14] for both utterance compression and meeting summarization. Each meeting is about an hour long. They are mainly research discussions on natural language processing, artificial intelligence, speech, and networking. All the meetings have been transcribed and annotated with dialogue acts (DAs) [15], topic boundaries, extractive and abstractive summaries [5]. 6 of the meetings are the commonly used test set for meeting summarization using the ICSI corpus ([5, 6, 8]), which contains 1088 extractive summary sentences from three annotators. The rest of the 20 meetings have only one summary annotation, with 1773 extractive summary sentences in total. We use the summary sentences from the 20 meetings as training data to develop the utterance compression module, then evaluate both the compression and summarization performance on the 6-meeting test set.

We use all the human annotated summary sentences from the 26 meetings (2861 summary sentences) for utterance compression annotation, conducted using the Amazon Mechanical Turk (AMT).[1] These sentences are grouped into 286 human intelligence tasks (HITs); each HIT contains 10 sentences that need to be compressed. Filled pauses such as "uh/um/eh" are removed in the preprocessing step to increase the sentence readability for human annotators.

We use a two-stage annotation scheme. In the first stage, each HIT was annotated by 8 mechanical turk workers. Each received \$0.15 as compensation for every HIT. For each sentence that needs to be compressed, the two sentences before and after it are displayed in the annotation interface in order to provide some context. We also show the speaker id for all the sentences since this is a multi-party conversation and knowing who said it is helpful to understand the utterance. The turkers can click on the unnecessary words and remove them from the original sentence; the resulting compressed sentence is shown in a preview text box. In total, 244 turkers participated in the first stage annotation. The average working time for compressing 10 sentences (one HIT) is 4.29 minutes.

In this study, for each utterance we only use one compression, the best compression found from the 8 annotations from the first stage. We use AMT again to conduct a second-stage annotation to find the best compression: we provide the same original summary sentence and its context to the annotators as in the first stage, list all the compression variants, and ask the turkers to select the best compression for each original summary sentence. Each sentence is annotated by 6 turkers in this annotation stage. Their majority vote is used as the gold standard compression. If there is a tie, we choose the shorter one. It takes 4 minutes on average for a turker to select the best compressions for 10 sentences. 300 turkers performed the second stage annotation. Only 41 of them are the same as in the first stage.[2] We found from the data that 16.12% of the selected best compressions are agreed on by all of the 6 annotators; 21.07% are agreed by 5 annotators; 28.70% are agreed by 4 annotators; 27.99% are agreed by 3 annotators; 6.09% are agreed by 2 annotators, and 0.03% by 1 annotator.

4. APPROACHES

4.1. Utterance Compression Modeling

Previous studies on sentence compression have been conducted mostly on written text domain. Popular approaches include the noisy-channel framework, integer linear programming, CRF model, tree transduction, and so on [16, 17, 18, 19, 20, 21]. In this work, we investigate sentence compression using spoken utterances. We follow [20] and formulate the spoken utterance compression task as a sequence labeling problem. We label a word with "0" if it is to be removed from the original utterance, and "1" if it is retained.

Given an original word sequence $X = (X_1, X_2, ..., X_n)$, the distribution of its corresponding label sequence $Y = (Y_1, Y_2, ..., Y_n)$ under the linear chain CRF model takes the following form:

$$p(Y|X) \propto \exp \sum_{k=1}^{n} (\sum_{j} \lambda_j f_j(y_k, y_{k-1}, X) + \sum_{i} \mu_i g_i(x_k, y_k, X))$$

where f_j are transition feature functions; g_i are observation feature functions; λ_j and μ_i are their corresponding weights.

[1] http://mturk.com

[2] We did not explicitly require different sets of turkers for these two stages. Given the large number of turkers in the two annotation stages and the number of HITs in our task, the chances that a turker works on the same set of utterances in the two stages (thus may be biased in the selection of best compressions in the second stage) are quite small.

We define the features g_i using a set of word tokens, part-of-speech tags, position features, and features extracted from the syntactic and discourse parsing trees.

- Word tokens
  This set of feature templates includes the identity of the current word token; the two word tokens before and after the current word; and all the bigrams and trigrams that can be formed by adjacent tokens and the current word.
- Part-of-Speech (POS) tags
  This set of feature templates includes the POS tags and the tag combinations that correspond to the unigram, bigram, and trigram word token features. We use the TnT part-of-speech tagger [22] trained from Switchboard data for tagging.
- Utterance length features
  There are two features: the length of the current utterance (measured by the total number of word tokens in it), and the relative position of the current word within the utterance (defined as the word position divided by the utterance length).
- Syntactic parsing tree based features
  We use the Charniak's reranking parser[3] to generate the sentence-level syntactic parsing tree for each utterance. We derive three types of feature templates from the syntactic parsing tree: (1) the second-to-last syntactic tag along the path from the root to the word, which denotes whether the current word token is included in the NP, VP, PP, ADVP phrases. We also include the same context tag information as defined for word token and POS feature templates. (2) length of the path starting from "S1" to the current word. (3) length of the path divided by the longest available path in the current parsing tree.
- Discourse parsing tree based features
  We use the sentence-level discourse parser "SPADE"[4] to generate the discourse parsing tree, whose leaves correspond to elementary discourse units and internal nodes correspond to discourse spans. We generate three types of feature templates from the discourse parsing tree: (1) length of the discourse unit containing the current word, measured by the number of word tokens in it. (2) relative position of the current word token within its discourse segment. (3) the first discourse tag ("Satellite" or "Nucleus") along the parsing tree.

We avoided deriving more complex features due to the concern that the POS tagger, syntactic and discourse parsers do not perform very well on the ill-formed spoken utterances.

4.2. Meeting Summarization

We choose to use the maximum marginal relevance (MMR) framework due to its simplicity and verified competency in speech summarization. We expect this is a good starting point for this study of spoken utterance compression for summarization. For each sentence S_i, its MMR score $MMR(S_i)$ is the linear combination of its similarity to the original document (or a user query), $Sim_1(S_i, D)$, and the similarity to the current selected summary sentences, $Sim_2(S_i, Summ)$, as shown below:

$$MMR(S_i) = \lambda \times Sim_1(S_i, D) - (1 - \lambda) \times Sim_2(S_i, Summ)$$

where λ is the balancing factor between the two components. We use cosine similarity under the vector space model for the similarity between two text segments (S_i and S_j):

$$Sim(S_i, S_j) = \frac{\sum_k w_{i,k} \times w_{j,k}}{\sqrt{\sum_k w_{i,k}^2} \times \sqrt{\sum_k w_{j,k}^2}}$$

The term weight for a word $w_{i,k}$ is determined by $TF \times IDF$, where TF is its term frequency in the text segment S_i, and IDF is the inverse document frequency generated from a large background corpus.

There are a variety of ways to combine the compression and summarization modules. In this study, we investigate using the compressed sentences (pre-compression) vs. the original transcripts (post-compression) as input for summarization.

- **Pre-compression**: we perform utterance compression on the original transcripts, then apply the MMR based summarization system on the compressed transcripts. For the summary output, we can use the selected compressed sentences, or map these sentences back to their corresponding original transcripts.
- **Post-compression**: we apply the MMR based summarization system on the original meeting transcripts. In this approach, to generate a compressed summary, there are two methods: (i) we can compress the selected summary sentences (these are in their original uncompressed format); (ii) we can simply map the selected summary sentences to their compressed version if sentences have been pre-compressed already.

We evaluate different configurations (in terms of MMR input and summary output) in order to answer the question whether using compressed sentences helps select better summary sentences for different summarization goals.

5. EXPERIMENTAL RESULTS

5.1. Compression Results

The first question we ask is, is it possible to automatically compress spoken utterances with reasonable performance? For this experiment, we use the CRF models for compression as described in Section 4.1. The CRF++[5] implementation was used. The training data is from the 20 training meetings, which contain 1,772 summary sentences (26,002 word tokens). The test data is from the 6-meeting test set, consisting of 1,088 sentences (16,361 word tokens). In order to generate output with different compression ratios,[6] we use the model's posterior probabilities. For every token, we calculate its confidence score: $p(keep) - p(delete)$, where $p(keep)$ and $p(delete)$ are the posterior probabilities of keeping and deleting this token respectively. We rank all the tokens on the test set according to this confidence measure and preserve tokens with high confidence scores until reaching the specified compression ratio. Note that we choose to use this corpus level compression ratio rather than at the sentence level, since different sentences may need different degrees of compression.

The utterance compression performance is evaluated using the token-level labeling accuracy and f-measure score, as well as the sentence-level labeling accuracy. The token-level accuracy is defined as the number of correctly labeled tokens divided by the total

[3] http://www.cs.brown.edu/~ec/#software

[4] http://www.isi.edu/licensed-sw/spade/

[5] http://crfpp.sourceforge.net/

[6] For an utterance, compression ratio is defined as the percentage of words preserved in the compressed utterance.

number of tokens in the test set. The f-score is the harmonic mean of precision and recall scores, using preserved tokens as the positive target class. The sentence-level labeling accuracy is the number of correctly compressed sentences divided by the total number of sentences in the test set. This is a more strict measure than the token level ones. Table 2 shows the utterance compression results using different compression ratios.

	token level				sent level
Ratio	P(%)	R(%)	F(%)	Acc(%)	Acc(%)
0.5	82.60	66.82	73.88	70.79	12.68
0.6	80.09	77.72	78.89	74.29	16.64
0.7	**77.21**	**87.41**	**81.99**	**76.27**	**19.85**
0.8	73.03	94.50	82.39	75.03	17.83
0.9	67.86	98.78	80.45	70.33	10.94

Table 2. Spoken utterance compression results on the test set using CRF models with different compression ratios.

We can see that a high compression ratio corresponds to high recall score and low precision, which is expected. When we retain 70% of the total words, the compression system achieved the best performance in terms of both accuracy and f-score, as well as the sentence level accuracy. This is also the compression ratio that is closest to that of the default CRF output, that is, the model determines whether a token is preserved or not without any given compression ratio. The compression ratio based on this default classifier output is 69.76%.

Regarding the features used in the CRF for sentence compression, we notice that the word identity and POS features are strong indicators of unnecessary words, and adding position related features and features extracted from syntactic and discourse parsing tree yielded slight performance improvement. Similar findings are also reported in [20].

5.2. Summarization Results

The second question we raise is, does compressing all the utterances in the transcripts before performing extractive meeting summarization help summary sentence selection? what is the best system setup for generating compressed extractive summaries? To answer these questions, we pre-compress the utterances in the original transcripts using the above CRF models with different word compression ratios. Then we apply the MMR approach for utterance selection using either pre-compressed transcripts or the original transcripts as input.[7] Summarization performance is evaluated using the widely adopted ROUGE metric [23].

In the first experiment, we evaluate the utterance selection performance of using both pre-compressed transcripts and original transcripts. When using the pre-compressed transcripts as MMR input, we map the selected summary sentences to their corresponding sentences in the original transcripts. The generated summaries are therefore compared against the human annotated extractive summaries. The summary length is set to be 15% of the total words in the original transcripts. This summarization ratio is similar to those used in previous work for meeting summarization. Results are shown in Table 3. Both ROUGE-1 and ROUGE-2 f-scores are presented to make our results comparable with previous studies. Higher ROUGE scores represent better utterance selection performance. We notice that when using compressed sentences in MMR with relatively high compression ratios (deleting limited words), there is moderate improvement, especially when measured by ROUGE-2 scores. We also evaluated using other summary length (shorter and longer), and found that in general, the difference in ROUGE-1 scores is rather small using the two different MMR inputs, and that ROUGE-2 results are slightly better using the compressed sentences in MMR.

[7]We used fixed parameter $\lambda = 0.5$ for all MMR experiments

	Ratio	R-1 F(%)	R-2 F(%)
MMR: Pre-compressed trans	0.5	70.19	36.00
	0.6	70.60	35.94
	0.7	71.07	**37.06**
	0.8	**71.21**	**37.75**
	0.9	**71.20**	**36.87**
MMR: Original trans		71.12	36.21

Table 3. Summarization results using both pre-compressed and original meeting transcripts. In both cases, selected sentences are mapped to the original sentences and compared against human annotated extractive summaries.

In the second experiment, we evaluate the performance of both pre-compression and post-compression in generating compressed meeting summaries. For pre-compression, we use the pre-compressed transcripts as input for the MMR system, then the top-ranked sentences are directly concatenated to form the compressed summaries until a pre-defined summary length is reached. For post-compression, we use the original transcripts as input for the MMR system, then map the top-ranked summary sentences to their corresponding compressed version to create the final compressed summary.

For this experiment, we use 10% summarization ratio (a smaller number than the previous experiment since the output is a compressed summary). All the generated summaries are compared against human compressed meeting summaries. ROUGE results are shown in Table 4. For a comparison, we also include results just using the original transcripts for summarization (with the same summary length, 10%), without any compression. We can see that both pre-compression and post-compression perform much better than simply using the original un-compressed extractive summaries (last row in Table 4). Between pre-compression and post-compression approaches, their difference is not very significant. The best results are achieved when using the pre-compression configuration, with different compression ratios for ROUGE-1 and ROUGE-2 measures. There is a larger improvement based on ROUGE-2 results than ROUGE-1 (bold numbers in the table).

As mentioned in Section 4.2, another alternative to perform post-compression is to compress the selected summary sentences, rather than mapping them to pre-compressed sentences. We also experimented with this setup. Take 70% utterance compression ratio as an example, we first selected important sentences containing 14.3% of the total words using the original transcripts as input, and then applied 70% compression ratio to these sentences, thus resulting in 14.3% $\times$ 70% $\approx$ 10% final summarization ratio. We found the results from this summarization-compression pipeline are slightly worse than those in Table 4, e.g., yielding 64.97 and 27.58 for R-1 and R-2 respectively for 70% compression ratio. The difference between the two ways of post-compression implementation lies in the definition of the utterance compression ratio, whether it is corpus based or at the sentence level. Further analysis is still needed to understand what is the best set up to generate compressed summaries.

Ratio	Pre-compression MMR: compressed sents		Post-compression MMR: original	
	R-1 F(%)	R-2 F(%)	R-1 F(%)	R-2 F(%)
0.5	62.78	25.06	63.55	25.07
0.6	64.29	**27.14**	64.83	26.37
0.7	**65.30**	**27.77**	65.01	27.27
0.8	64.95	**28.26**	65.09	27.50
0.9	**64.26**	**26.95**	64.03	26.13
Orig trans	R-1 F(%): 60.38		R-2 F(%): 22.86	

Table 4. Summarization results using pre-compression, post-compression, and no-compression (extractive summary sentences are rendered using original transcripts). The generated summaries are compared against human compressed meeting summaries.

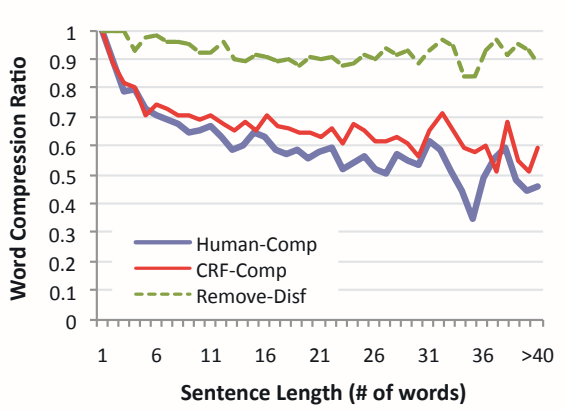

Fig. 1. Average sentence level word compression ratio with respect to different sentence length.

Our experimental results indicate that overall using compressed sentences for summarization performs similarly or slightly better for both summarization goals: generating original extractive summaries and compressed ones. It is also worth mentioning that when summary length is prespecified (in terms of number of words), using compressed summaries allows the system to include more sentences in the summary, increasing the coverage of important content. This is especially true for conversational speech, in which many words can be removed without significantly affecting information content. The generated compressed summaries improve readability and are closer to abstractive summaries.

5.3. Comparison with Disfluency Removal

We have performed experiments using different compression ratios above. When compression ratios are high, only a small percentage of words are removed. We expect that those words are likely to be disfluencies. The human annotators typically first remove repetitions, revisions, and other disfluencies to simplify the sentence, then remove other words/phrases to further compress it. In this section, we compare sentence compression with disfluency removal for summarization. [24] used human annotated disfluencies and evaluated the effect of disfluency removal on meeting summarization. They showed that removing disfluencies first, followed by MMR extractive summarization, did not help summary sentence selection. Our results (as shown in Table 3 when compared to human extractive summaries) are similar, especially when measured using ROUGE-1 F-scores, but we observed some improvement using compressed sentences for summarization based on ROUGE-2 results.

Liu et al. [24] used the same 6-meeting test set, therefore we can compare their disfluency annotation with our utterance compression annotations on the same summary sentences of these 6 meetings. The corpus-level word compression ratio is 91.58% for disfluency cleaned-up data, and 58.12% for our compression data.[8] Figure 1 shows the average sentence level word compression ratio with respect to different sentence length for different data: using human compression vs. disfluency removal. We also include the analysis for the automatic compression output using the CRF model (with 70% compression ratio) in the figure for a comparison. We notice that the word compression ratios for disfluency cleaned-up data do not change much for sentences with different length; while for the gold standard compression results, there is a clear trend that longer utterances tend to be compressed more. The automatic compression output shows similar tendency as human compressions – compressing longer utterances more aggressively. This relationship between sentence length and compression ratio also explains why we use corpus level compression ratio instead of sentence level for automatic utterance compression. The above analysis was conducted using the human disfluency annotation results. In the future, we will compare the automatic generated output of a disfluency removal module and an utterance compression module.

[8]Note that these word compression scores are calculated with respect to the original transcripts, while our word compression ratios used for the CRF models were calculated based on the pre-cleaned data (removing "uh/um/eh" for annotation purpose). There is about 3.5% difference between these two measures.

6. CONCLUSION

In this paper, we proposed to automatically generate compressed meeting summaries and improve summarization quality. We modeled the utterance compression task as a sequence labeling problem, and showed satisfying performance using a CRF model that incorporates word identity, part-of-speech, position, and features extracted from syntactic and discourse parsing tree. This utterance compression module was combined with an MMR based extractive meeting summarization system. We compared using different sentence inputs in MMR: original vs. compressed sentences, and found that the latter performs slightly better, but the difference between the two setups is rather small. Overall, we demonstrated that spoken utterance compression is feasible and that we can generate compressed summaries with reasonable performance. This is one step towards automatic abstractive summarization. In addition, another important contribution of this work is the corpus of compressed spoken utterances we created, which can be used for cross-genre studies.

The compression approach we used in this study can be improved in many ways. For example, we did not explicitly consider the sentence structure. In the future, we may first generate a set of syntactically well-formed utterances, then select the best compression from them. We can also experiment with incorporating prosodic features in the CRF models. Other than the 1-best gold standard compression we used in the experiments, we may consider other alternative compressions. For summarization, we will investigate the possibilities of jointly optimizing the compression and summarization systems. It is possible to let the summarization system decide whether to use the compressed or the original sentences [25],

or whether a particular word should be removed or not in order to generate a high-quality summary. Furthermore, we also plan to investigate cross-sentence fusion to generate more coherent abstract-alike summaries.

We used automatic evaluation in this paper: word or sentence level accuracy for compression, and ROUGE for summarization. We plan to perform more human evaluation for these two modules. For compression, we will evaluate compressed utterances based on grammaticality and informativeness, as conducted in [12]. For summarization, [26] showed that ROUGE scores only measure content match and do not correlate well with human evaluations for some aspects of summary quality. In our future studies, we will perform human readability test on the compressed summaries, and evaluate information content and overall quality of the summaries.

7. ACKNOWLEDGMENTS

We thank the three anonymous reviewers for their valuable suggestions. We also thank Dr. Fuliang Weng and Dr. Liang Huang for useful discussions. This work is supported by NSF award IIS-0845484. Any opinions expressed in this work are those of the authors and do not necessarily reflect the views of NSF.

8. REFERENCES

[1] K. Zechner, "Automatic summarization of open-domain multiparty dialogues in diverse genres," *Computational Linguistics*, vol. 28, no. 4, pp. 447–485, 2002.

[2] D. Gillick, K. Riedhammer, B. Favre, and D. Hakkani-Tür, "A global optimization framework for meeting summarization," in *Proc. of ICASSP*, 2009.

[3] S. Xie, B. Favre, D. Hakkani-Tur, and Y. Liu, "Leveraging sentence weights in a concept-based optimization framework for extractive meeting summarization," in *Proc. of INTERSPEECH*, 2009.

[4] S. Maskey and J. Hirschberg, "Comparing lexical, acoustic/prosodic, structural and discourse features for speech summarization," in *Proc. of Eurospeech*, 2005.

[5] G. Murray, S. Renals, J. Carletta, and J. Moore, "Evaluating automatic summaries of meeting recordings," in *Proceedings of ACL 2005 MTSE Workshop*, 2005, pp. 39–52.

[6] M. Galley, "A skip-chain conditional random field for ranking meeting utterances by importance," in *Proc. of EMNLP*, 2006.

[7] J. Zhang, H. Y. Chan, P. Fung, and L. Cuo, "A comparative study on speech summarization of broadcast news and lecture speech," in *Proc. of Interspeech*, 2007.

[8] S. Xie, D. Hakkani-Tur, B. Favre, and Y. Liu, "Integrating prosodic features in extractive meeting summarization," in *Proc. of ASRU*, 2009.

[9] S.-H. Lin and B. Chen, "A risk minimization framework for extractive speech summarization," in *Proc. of ACL*, 2010.

[10] C. Hori and S. Furui, "A new approach to automatic speech summarization," *IEEE Transactions on Multimedia*, vol. 5, no. 3, pp. 368–378, 2003.

[11] G. Murray, G. Carenini, and R. Ng, "Interpretation and transformation for abstracting conversations," in *Proc. of NAACL*, 2010.

[12] F. Liu and Y. Liu, "From extractive to abstractive meeting summaries: Can it be done by sentence compression?," in *Proc. of ACL*, 2009.

[13] A. Janin, D. Baron, J. Edwards, D. Ellis, G. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI meeting corpus," in *Proceedings of ICASSP*, 2003, pp. 364–367.

[14] F. Liu, F. Liu, and Y. Liu, "A supervised framework for keyword extraction from meeting transcripts," *IEEE Trans. on Audio, Speech, and Language Processing*, accepted.

[15] E. Shriberg, R. Dhillon, S. Bhagat, J. Ang, and H. Carvey, "The ICSI meeting recorder dialog act (MRDA) corpus," in *Proceedings of SIGdial Workshop on Discourse and Dialogue*, 2004, pp. 97–100.

[16] K. Knight and D. Marcu, "Summarization beyond sentence extraction: A probabilistic approach to sentence compression," *Artificial Intelligence*, vol. 139, pp. 91–107, 2002.

[17] J. Turner and E. Charniak, "Supervised and unsupervised learning for sentence compression," in *Proc. of ACL*, 2005.

[18] M. Galley and K. McKeown, "Lexicalized markov grammars for sentence compression," in *Proc. of NAACL/HLT*, 2007.

[19] J. Clarke and M. Lapata, "Global inference for sentence compression: An integer linear programming approach," *Journal of Artificial Intelligence Research*, vol. 31, pp. 399–429, 2008.

[20] T. Nomoto, "Discriminative sentence compression with conditional random fields," *Information Processing and Management*, vol. 43, pp. 1571 – 1587, 2007.

[21] T. Cohn and M. Lapata, "Sentence compression as tree transduction," *Journal of Artificial Intelligence Research*, vol. 34, pp. 637–674, 2009.

[22] T. Brants, "TnT – a statistical part-of-speech tagger," in *Proceedings of the 6th Applied NLP Conference*, 2000, pp. 224–231.

[23] C. Lin, "ROUGE: A package for automatic evaluation of summaries," in *Proceedings of ACL Workshop on Text Summarization Branches Out*, 2004, pp. 74–81.

[24] Y. Liu, F. Liu, B. Li, and S. Xie, "Do disfluencies affect meeting summarization? a pilot study on the impact of disfluencies," in *Proc. of MLMI*, 2007.

[25] D. M. Zajic, J. Lin, B. Dorr, and R. Schwartz, "Sentence compression as a component of a multi-document summarization system," in *Proc. of DUC*, 2006.

[26] F. Liu and Y. Liu, "Exploring correlation between rouge and human evaluation on meeting summaries," *IEEE Trans. on Audio, Speech, and Language Processing*, vol. 18, no. 1, pp. 187–196, 2010.