UNSUPERVISED APPROACH TO EXTRACT SUMMARY KEYWORDS IN MEETING DOMAIN

Mohammad Hadi Bokaei[†,‡], *Hossein Sameti*[†], *Yang Liu*[‡]

Department of Computer Engineering, Sharif University of Technology, Tehran, Iran [†]
Department of Computer Science, The University of Texas at Dallas, U.S.A [‡]

ABSTRACT

Summary keywords are words that are used in the reference extracted summary, therefore can be used to discriminate between summary sentences from non-summary ones. Finding these words is important for the extractive summarization algorithms that measure the importance of a sentence based on the importance of its constituent words. This paper is focused on extracting summary keywords in the multi-party meeting domain. We test previously proposed keyword extraction algorithms and evaluate their performance to determine summary keywords. We also propose a new approach which uses discourse information to find local important keywords and show that it outperforms all the previous methods. We evaluate our proposed approach on the standard AMI meeting corpus according to the reference extracted summary prepared in this corpus.

Index Terms— meeting segmentation, funtion segmentation, unsupervised algorithm, summary keyword extraction.

1. INTRODUCTION

Meeting summarization becomes an attractive research trend in recent years. Various methods have been proposed to find the most important utterances within a meeting transcript, such as Maximal Marginal Relevance (MMR) [1], topic-based [2, 3], graph-based [4, 5], and optimization-based [6, 7] methods. Most of these approaches rely on measuring the importance of the constituent words of each utterance. For example concept-based Integer Linear Programming (ILP) summarizer [6] needs to find important words (known as concepts) to use them in an optimization framework to extract the best combination of utterances. Apparently the success of these methods is highly dependent on the quality of the found concepts.

summary keywords [8] are the words (or phrases) which have a good capability to discriminate between summary and non-summary utterances. According to this definition, in a meeting about "designing a remote control", the phrase "remote control" is a keyword, but it is not a summary indicative word, because it is used both in summary and non summary utterances. Instead words like "meeting", "decision", and "attention" are examples of summary keywords, since they are used mostly in the utterances which are marked as important by the annotators. This paper focuses on finding these words in a given meeting transcript using an unsupervised approach.

The closest trend to this topic is keyword extraction. A lot of work has been done on this topic in various domains like paper abstracts [9], blogs [10], news articles [11], course lectures [12], and meetings [13]. Among these trends, extracting keywords from meeting transcripts is a more challenging task, because of specific characteristics like low lexical density, lack of structural information, multiple participants with different styles of talking, and spontaneous speech which causes high recognition error rate and disfluencies in the meeting transcript [13].

It has been previously shown that the most successful unsupervised keyword extraction approaches in the meeting domain are Term Frequency Inverse Document Frequency (TFIDF) and graph-based methods [14, 15]. In both of these approaches, words are scored according to a specific criterion and the top best words are returned as the keywords. In TFIDF each word w is scored based on its frequency in the document and how many other documents include w. In graph-based methods (like TextRank [16] and SingleRank [11]), a graph is constructed in which words are vertices and edges are weighted according to specific features of the corresponding words. Inspired by Google's PageRank algorithm [17], a random-walk procedure is applied to the constructed graph to compute the final score for each word.

There has been little work directly focused on finding summary keywords. The authors in [18] showed that using reference summary keywords in ILP framework can improve the accuracy of the extractive summarization system. They proposed a method which estimates bigram frequency in the reference summary and used this estimation to improve ILP summarization of documents. A supervised regression-based approach was also introduced in [8] in which an importance weight is assigned to each word according to various kinds of features. They showed that these weights can be used to improve multi-document extractive summarization performance. However these approaches are supervised and need an annotated set to train the underlying model.

To the best of our knowledge, there is no previous work

978-0-9928626-3-3/15/$31.00 ©2015 IEEE

which tackles the problem of finding summary keywords in meeting domain. The contribution of this paper is as follows: 1) We evaluate the most accurate previously proposed unsupervised keyword extraction algorithms to find the summary keywords in the meeting domain. 2) We propose a new approach in which discourse information is also considered to calculate word score and show that it outperforms all other base-line algorithms.

2. PROPOSED METHOD

The goal of our ongoing research trend is to explore meeting discourse information and use it for the summarization task. As the first step we propose an unsupervised algorithm which aims to segment a meeting transcript into shorter parts, each one representing an event in the meeting [19]. This kind of segmentation is called function segmentation [20]. Our target categories to segment a meeting are: $Monologue_i$ and $Discussion_{x_1 \ldots x_n}$, where i can be any one of participants in the meeting and x_i is a binary value which denotes whether speaker i is involved in the discussion or not. In this section, we first introduce our approach to segment a meeting and then show how we use this segmentation to find summary keywords within the meeting.

2.1. Discourse segmentation approach

The main idea in our segmentation approach is that the involved participant set is not changed significantly during the course of one function segment. In contrast there must be a noticeable change in this set between two segments. We try to find these points in the input sequence.

The input to our segmentation algorithm is the transcript of the meeting's utterances as well as the speaker of each utterance. Since the main usage of this algorithm can be considered as the first step of other meeting understanding tasks, the transcript, the boundaries of utterances and the speaker of each utterance is assumed to be prepared using an automatic speech recognition engine[1].

All the elements in the input sequence are considered to be a possible boundary for the segmentation. For each possible boundary at position *i*, two windows are placed over elements $[i - L_w : i - 1]$ (win_L^i) and $[i : i + L_w - 1]$ (win_R^i). L_w is the window length and will be tuned according to our development set. From each window, two sets of features are extracted to be used to find the boundary points.

In the first one, contribution of each participant is measured in terms of their uttered words:

$$FV_1^{win}(i) = \frac{num\_word_i^{win}}{\sum_j num\_word_j^{win}} \quad (1)$$

where $num\_word_i^{win}$ is the total number of words uttered by the participant i in the window win.

[1]These are common assumptions in most meeting understanding tasks.

In the second set of features, the importance of words uttered by each participant is measured according to the total amount of $tfidf$ scores. The formula for the second group of features is:

$$FV_2^{win}(i) = \frac{\sum_{w \in Words_i^{win}} tfidf(w)}{\sum_j \sum_{w \in Words_j^{win}} tfidf(w)} \quad (2)$$

where $Words_i^{win}$ is the total words which are uttered by participant i in the window win.

The score of the possible boundary i, $sc(i)$, is considered to be the distance between the feature vectors extracted from win_L^i and win_R^i according to Jeffrey divergence [21], which is a numerically stable and symmetric form of the Kullback-Leibler distance metric [22]:

$$sc(i) = \sum_{j=1}^{2} Jef(FV_j^{win_L^i}, FV_j^{win_R^i}) \quad (3)$$

where $Jef(p, q)$ is:

$$\sum_i p(i).log(\frac{p(i)}{\frac{p(i)+q(i)}{2}}) + q(i).log(\frac{q(i)}{\frac{p(i)+q(i)}{2}}) \quad (4)$$

where $p(i)$ and $q(i)$ are the ith elements of the vectors p and q respectively.

These scores are computed for each possible boundary to form the score plot. This plot is smoothed according to Equation 5:

$$sc(i) = \frac{1}{s+1} \sum_{j=i-s/2}^{i+s/2} sc(j) \quad (5)$$

where s is the parameter of the smoothing algorithm. The $peak\_score$ is then obtained for each position according to Equation 6:

$$peak\_score(cp) = 2sc(cp) - sc(pw) - sc(nw) \quad (6)$$

where cp is the point at which we want to calculate the $peak\_score$, and pw and nw are the locations of the nearest valleys to cp on its left and right side respectively.

Finally we return the boundaries with the highest $peak_score$s. Since the number of segments is not known in advance, we calculate the average ($\bar{s}$) and the standard deviation (σ) of all the calculated $peak_score$s and return candidate boundaries whose $peak_score$s are higher than $\bar{s}-\sigma$.

2.2. Extracting summary keywords

In order to extract the summary keywords, we first filter out unnecessary words from the input transcript. We use two common heuristics [14]: (1) using a stop word list to remove unimportant words and (2) allowing words with specific part of speech tags (POS) to be considered as candidate words[2].

[2]Using Stanford POS tagger [23], we consider nouns (N, NN, NNP, NNPS, NNS), adjectives (JJ, JJR, JJS), and verbs (VB, VBD, VBG, VBN, VBP, VBZ) as the candidate words.

We then compute a score for each candidate word. This score is calculated according to two different criteria: global score which is the IDF score of the word according to all the documents in the corpus and is computed according to Equation 7:

$$idf(w) = log(\frac{D_r}{D(w)}) \tag{7}$$

where D_r is the total number of documents in the corpus and $D(w)$ is the number of documents in which word w is used.

We additionally use a local score for each word which measures the informativeness of the word according to the entropy of the word's usage among segments in the found segmentation (Section 2.1). We first define $p(s_j|w)$ to be the probability of being in segment s_j, if the word w is observed:

$$p(s_j|w) = \frac{n(w, s_j)}{\sum_k n(w, s_k)} \tag{8}$$

where $n(w, s_j)$ is the number of times that the word w is used in the segment s_j. According to this probability, $nent(w, S)$ is the negated of the entropy of the word w:

$$nent(w, S) = \sum_{\{s_j|w \in s_j\}} p(s_j|w) \log p(s_j|w) \tag{9}$$

where the summation is taken over the segments that contain the word w. If the distribution $p(s_j|w)$ is flat for a specific word, meaning that the word is used evenly in all segments, that word is probably not an informative word for a specific segment and its $nent(w, S)$ score is a low value. On the other hand, if a word is used just in a few segments, the distribution $p(s_j|w)$ has peaks on those segments and the $nent(w, S)$ is a high value.

The total score for each word is computed by simply adding both of the global and local scores (Equations 7 and 9). These words are then sorted according to their scores and the top $T\%$ of all unique words are returned as summary keywords. In Section 3 we will evaluate the performance of the algorithm according to various choices of T.

3. EXPERIMENTAL RESULTS

The AMI meeting corpus [24] is a collection of 100 hours of meeting data and includes annotations in various layers such as speech audio, transcripts, focus of attention, etc. For each meeting, maximum of three reference extractive summaries are prepared which are used to evaluate our proposed algorithm. Automatic transcripts are provided by the AMI ASR team [25], yielding a word error rate (WER) of about 36%.

In order to evaluate the effectiveness of the proposed segmentation algorithm, a subset of 11 meetings in the AMI corpus are manually annotated and used as our test set[3]. We employed graduate students as annotators and asked them to segment the meetings according to different events. They were given a guideline which included the task definition and various examples used to clarify the concept of events and function segmentation of a meeting. Each meeting was annotated by one annotator. We used one meeting *is1003b* as our development set on which we tuned the parameter of our proposed segmentation algorithm (L_w).

Table 1. Results of our proposed segmentation algorithm.

Algorithm	$P_k(\%)$
Random segmentation	50.00
Proposed segmentation algorithm	26.93

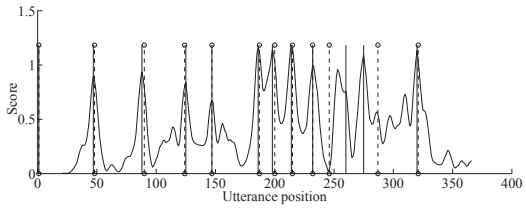

Fig. 1. Results of applying the segmentation algorithm on the sample meeting *es2008a*. Dashed lines and solid lines show the location of found and reference boundaries respectively.

We use P_k [26] as the evaluation metric which is a measure of error and thus a lower score means better segmentation performance. The formula for P_k is shown in Equation 10.

$$P_k = \frac{\sum_{i=1}^{N-k} \delta_H(i, i+k) \oplus \delta_R(i, i+k)}{N-k} \tag{10}$$

where H is the system generated segmentation and R is the reference segmentation. Given a segmentation S, $\delta_S(i, j)$ is a function which outputs 1 if and only if the segmentation S assigns ith and jth element to the same segment. The choice of k is arbitrary, but is generally set to be half of the average segment length in the reference segmentation.

We empirically choose L_w (window length) to be 20. As recommended in previous work [27], we also choose s (smoothing window length) to be 5. Table 1 shows the results of our proposed segmentation algorithm and compares it with random segmentation. An example of applying the algorithm on a sample meeting (*es2008a*) is also shown in Figure 1. In this figure the score plot (Equation 5) is also shown for each utterance boundary. Results show that most of the reference boundaries are found successfully using the proposed segmentation algorithm.

In order to evaluate the whole summary keyword detection algorithm, we use all the 134 meetings in the AMI cor-

[3]The ids of annotated meetings are: *es2008a*, *is1000a*, *is1001b*, *is1001c*, *is1006b*, *is1007b*, *is1008a*, *is1008b*, *is1008c*, *ts3005a* and *is1003b*. We chose these meetings since they have more annotations (such as focus of attention and addressee information) in the AMI corpus, which can be useful for our future studies. The reference segmentation as well as the annotation guide can be found here: http://ce.sharif.edu/ bokaei/resources/funseg/

Table 2. Results of our proposed algorithm to find the summary keywords, compared to other base-line algorithms.

T(%)	Algorithm	P(%)	R(%)	F(%)
10	TextRank	2.86	1.68	2.02
	SingleRank	7.07	4.29	5.12
	TFIDF	13.68	7.89	9.64
	IDF	28.26	17.54	20.83
	Proposed	**29.14**	**18.17**	**21.54**
20	TextRank	7.30	8.32	7.45
	SingleRank	10.30	12.16	10.69
	TFIDF	18.42	22.02	19.28
	IDF	28.46	35.38	30.32
	Proposed	**29.68**	**36.74**	**30.47**
30	TextRank	13.49	23.96	16.63
	SingleRank	14.09	25.38	17.44
	TFIDF	20.57	37.27	25.55
	IDF	28.56	53.34	35.86
	Proposed	**29.90**	**55.97**	**37.45**

pus. For each meeting the prepared reference extracted summaries are first preprocessed (using the same two rules explained in Section 2.2) and then words that are used just in reference summaries are determined. We finally come up with separate reference keyword sets for each meeting which are used to evaluate our proposed algorithm. The average number of the summary keywords in the AMI corpus is 101. We use well-known precision, recall and f-measure to compare the extracted summary keywords against each reference set for each meeting in the test set.

Results of the keyword extraction algorithm are shown in Table 2. We compare our proposed algorithm with the best previous keyword extraction algorithms. For all methods, we use a preprocessing step in which the candidate words are selected (as explained in Section 2.2). From these results, it can be seen that the graph-based methods (TextRank and SingleRank) are not accurate for this task. Consistent with previous work [14], TFIDF has better performance than the graph-based methods. However it can be seen that when we just use IDF score, the results become much better. The main reason is that a word with high TFIDF score is more likely to have high term frequency, which increase its chance to occur in a non-summary sentences and accordingly is not selected in the reference summary keyword set.

When we add local measures (proposed method) we have additional improvement over IDF approach to find the summary keywords. The main reason of this improvement is due to the capability of this new measure to highlight words that are important in a specific portion of the transcript. These words are important in just a few segments. Our local scoring measure (Equation 9) tries to capture these locally important words.

4. CONCLUSION AND FUTURE WORK

In this paper we focused on extracting summary keywords. We evaluated previous keyword extraction algorithms for this new task and proposed a new approach which outperforms all previous ones. This work is our first step toward our ultimate goal of summarization using discourse information. For our future work we aim to further improve the summary keyword extraction algorithm and then use the keywords to improve the state-of-the-art summarization methods.

5. ACKNOWLEDGMENT

This work is partly supported by NSF award IIS-0845484. Any opinions expressed in this material are those of the authors and do not necessarily reflect the views of NSF.

REFERENCES

[1] Shasha Xie and Yang Liu, "Using corpus and knowledge-based similarity measure in maximum marginal relevance for meeting summarization," in *proc. of International Conference on Acoustic, Speech, and Signal Processing (ICASSP), Las Vegas, NV*, 2008, pp. 4985–4988.

[2] Gabriel Murray, Steve Renals, Jean Carletta, and Johanna Moore, "Evaluating automatic summaries of meeting recordings," in *proc. of ACL MTSE Workshop, Ann Arbor, MI, USA*, 2005, pp. 33–40.

[3] Sheng-Yi Kong and Lin-shan Lee, "Improved spoken document summarization using probabilistic latent semantic analysis (PLSA)," in *proc. of International Conference on Acoustic, Speech, and Signal Processing (ICASSP), Toulouse*, 2006, vol. 1, pp. 941–944.

[4] Nikhil Garg, Benoit Favre, Korbinian Riedhammer, and Dilek Hakkani-Tür, "Clusterrank: a graph based method for meeting summarization," in *proc. of 10th Annual Conference of the International Speech Communication Association (INTERSPEECH), Brighton, UK*, 2009, pp. 1499–1502.

[5] Yun-Nung Chen and Florian Metze, "Multi-layer mutually reinforced random walk with hidden parameters for improved multi-party meeting summarization.," in *proc. of Annual Conference of the International Speech Communication Association (INTERSPEECH), Lyon, France*, 2013, pp. 485–489.

[6] Daniel Gillick, Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-Tur, "A global optimization framework for meeting summarization," in *proc. of International Conference on Acoustic, Speech, and Signal Processing (ICASSP), Taipei*, 2009, pp. 4769–4772.

[7] Shasha Xie, Benoit Favre, Dilek Hakkani-Tür, and Yang Liu, "Leveraging sentence weights in a concept-

based optimization framework for extractive meeting summarization.," in *proc. of 10th Conference of the International Speech Communication Association (INTERSPEECH), Brighton, UK*, 2009, pp. 1503–1506.

[8] Kai Hong and Ani Nenkova, "Improving the estimation of word importance for news multi-document summarization," in *proc. of the 14th Conference of the European Chapter of the Association for Computational Linguistics, Gothenburg, Sweden*, 2014, pp. 712–721.

[9] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun, "Clustering to find exemplar terms for keyphrase extraction," in *proc. of the 2009 Conference on Empirical Methods in Natural Language Processing, Singapore*, 2009, pp. 257–266.

[10] Maria Grineva, Maxim Grinev, and Dmitry Lizorkin, "Extracting key terms from noisy and multitheme documents," in *proc. of the 18th international conference on World wide web, Madrid*, 2009, pp. 661–670.

[11] Xiaojun Wan and Jianguo Xiao, "Single document keyphrase extraction using neighborhood knowledge," in *proc. of the 23rd AAAI Conference on Artificial Intelligence, Chicago, US*, 2008, pp. 855–860.

[12] Yun-Nung Chen, Yu Huang, Sheng-Yi Kong, and Lin-Shan Lee, "Automatic key term extraction from spoken course lectures using branching entropy and prosodic/semantic features," in *proc. of Spoken Language Technology Workshop (SLT), Berkeley, CA*, 2010, pp. 265–270.

[13] Fei Liu, Feifan Liu, and Yang Liu, "A supervised framework for keyword extraction from meeting transcripts," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 538–548, 2011.

[14] Kazi Saidul Hasan and Vincent Ng, "Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art," in *proc. of the 23rd International Conference on Computational Linguistics, Stroudsburg, PA*, 2010, pp. 365–373.

[15] Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu, "Unsupervised approaches for automatic keyword extraction using meeting transcripts," in *proc. of the annual conference of the North American chapter of the Association for Computational Linguistics, Boulder, Colorado*, 2009, pp. 620–628.

[16] Rada Mihalcea and Paul Tarau, "Textrank: Bringing order into texts," in *proc. of the Conference on Empirical Methods in Natural Language Processing, Barcelona, Spain*, 2004, pp. 404–411.

[17] Sergey Brin and Lawrence Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1, pp. 107–117, 1998.

[18] Chen Li, Xian Qian, and Yang Liu, "Using supervised bigram-based ILP for extractive summarization.," in *proc. of Annual meeting of the Association for Computational Linguistics, Sofia, Bulgaria,*, 2013, pp. 1004–1013.

[19] Iain McCowan, Samy Bengio, Daniel Gatica-Perez, Guillaume Lathoud, Florent Monay, Darren Moore, Pierre Wellner, and Hervé Bourlard, "Modeling human interaction in meetings," in *proc. of IEEE International conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003, vol. 4, pp. 748–751.

[20] Bonnie Webber, Markus Egg, and Valia Kordoni, "Discourse structure and language technology," *Journal of Natural Language Engineering*, vol. 18, no. 4, pp. 437–490, 2012.

[21] Jan Puzicha, Thomas Hofmann, and Joachim M Buhmann, "Non-parametric similarity measures for unsupervised texture segmentation and image retrieval," in *proc. of Computer Vision and Pattern Recognition, San Juan*, 1997, pp. 267–272.

[22] Solomon Kullback and Richard A Leibler, "On information and sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[23] Kristina Toutanova and Christopher D Manning, "Enriching the knowledge sources used in a maximum entropy part-of-speech tagger," in *proc. of the conference on Empirical methods in natural language processing*, 2000, pp. 63–70.

[24] Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Mael Guillemot, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, et al., "The AMI meeting corpus: A pre-announcement," in *proc. of Machine Learning for Multimodal Interaction, Edinburgh, UK*, 2006, pp. 28–39.

[25] Steve Renals, Thomas Hain, and Hervé Bourlard, "Recognition and understanding of meetings the ami and amida projects," in *proc. of the IEEE workshop on Recognition and understanding of meetings the AMI and AMIDA projects, Kyoto, Japan*, 2007, pp. 238–247.

[26] Doug Beeferman, Adam Berger, and John Lafferty, "Statistical models for text segmentation," *Machine learning*, vol. 34, no. 1-3, pp. 177–210, 1999.

[27] Marti A Hearst, "Texttiling: Segmenting text into multi-paragraph subtopic passages," *Computational linguistics*, vol. 23, no. 1, pp. 33–64, 1997.