

# Abstractive Meeting Summarization Using Dependency Graph Fusion

Siddhartha Banerjee  
The Pennsylvania State  
University  
University Park  
PA, USA - 16802  
sub253@ist.psu.edu

Prasenjit Mitra  
Qatar Computing Research  
Institute  
Tornado Tower, 18th floor  
Doha, Qatar  
pmitra@qf.org.qa

Kazunari Sugiyama  
National University of  
Singapore  
13 Computing Drive  
Singapore 117417  
sugiyama@comp.nus.edu.sg

## ABSTRACT

Automatic summarization techniques on meeting conversations developed so far have been primarily extractive, resulting in poor summaries. To improve this, we propose an approach to generate abstractive summaries by fusing important content from several utterances. Any meeting is generally comprised of several discussion topic segments. For each topic segment within a meeting conversation, we aim to generate a one sentence summary from the most important utterances using an *integer linear programming*-based sentence fusion approach. Experimental results show that our method can generate more informative summaries than the baselines.

## Categories and Subject Descriptors

I.2.7 [Artificial Intelligence]: Natural Language Processing—*Language generation*

## Keywords

Abstractive meeting summarization; Integer linear programming

## 1. INTRODUCTION

Meeting summarization helps both participants and non-participants by providing a short and concise snapshot of the most important content discussed in the meetings. A recent study revealed that people generally prefer abstractive summaries [4]. Table 1 shows the human-written abstractive summaries along with the human-generated extractive summaries from a meeting transcript. As can be seen, the utterances are highly noisy and contain unnecessary information. Even if an extractive summarizer can accurately classify these utterances as “important” and present them to a reader, it is hard to read and synthesize information from such utterances. In contrast, human written summaries are compact and readable.

We propose an automatic way of generating short and concise abstractive summaries of meetings. Any meeting conversation includes dialogues on several topics. For example, in Table 1, the participants converse on two topics: *design features* and *selling prices*. Given the most important sentences within a topic segment, our goal is to generate a one-sentence summary from each segment and appending them to form a comprehensive summary of the meeting. Moreover, we also aim to generate summaries that resemble human-written summaries in terms of writing style.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s). Copyright is held by the author/owner(s).  
WWW 2015 Companion, May 18–22, 2015, Florence, Italy.  
ACM 978-1-4503-3473-0/15/05.  
<http://dx.doi.org/10.1145/2740908.2742751>.

**Table 1: Two sets of extractive summaries and gold standard human generated abstractive summaries from a meeting (Set 2 follows Set 1).**

<b>Set 1: Human-generated extractive summary</b> D: um as well as uh characters D: um different uh keypad styles and s symbols. A: Well right away I'm wondering if there's um th th uh, like with D_V_D players, if there are zones. A: Cause you have more complicated characters like European languages, then you need more buttons. D: I'm thinking the price might appeal to a certain market in one region, whereas in another it'll be different, so D: kay trendy probably means something other than just basic <b>Abstractive summary:</b> The team then discussed various features to consider in making the remote.
<b>Set 2: Human-generated extractive summary</b> B: Like how much does, you know, a remote control cost. B: Well twenty five Euro, I mean that's um that's about like eighteen pounds or something. D: This is this gonna be to be like the premium product kinda thing or B: So I don't know how good a remote control that would get you. Um. <b>Abstractive summary:</b> The project manager talked about the project finances and selling prices.

To aggregate the information from multiple utterances, we adapt an existing integer linear programming (ILP) based fusion technique [1]. The fusion technique is based on the idea of merging dependency parse trees of the utterances. The trees are merged on the common nodes that are represented by the word and parts-of-speech (POS) combination. Each edge of the merged structure is represented as a variable in the ILP objective function, and the solution will decide whether the edge has to be preserved or discarded. We modify the technique by introducing an anaphora resolution step and also an ambiguity resolver that takes the context of words into account. Further, to solve the ILP, we introduce several constraints, such as desired length of the output, etc.

To the best of our knowledge, our work is the first to address the problems of readability, grammaticality and content selection jointly for meeting summary generation without employing a template-based approach. We conduct experiments on the AMI corpus<sup>1</sup> that consists of meeting transcripts and show that our best method outperforms extractive model significantly on ROUGE-2 scores (0.048 vs 0.026).

## 2. PROPOSED APPROACH

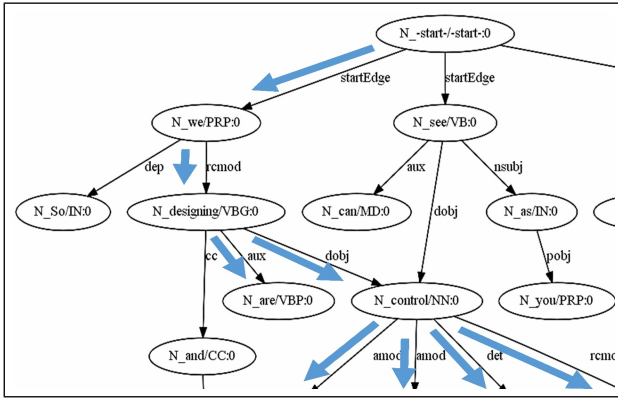
Dependency fusion on meeting data requires an algorithm that is robust for noisy data as utterances often have disfluencies. Our work applies fusion to all the important utterances within the topic segment to generate the best sub-tree that satisfies the constraints and maximizes the objective function of the optimization problem. Anaphora resolution step replaces pronouns with the original nouns in the previous utterance that they refer to in order to increase the chances of merging. Consider the following utterances:

“so we're designing a new remote control and um”

“Um, as you can see it's supposed to be original”

Without pronoun resolution, these two utterances cannot be merged. Once we apply anaphora resolution, *it* in the second utterance is modified to a *new remote control* and then both the utterances are fused into a common structure. The utterances are parsed using the Stanford dependency parser. Every individual utterance has an explicit ROOT node. We add two dummy nodes in the graph – the

<sup>1</sup><http://groups.inf.ed.ac.uk/ami/corpus/>



**Figure 1: A merged dependency graph structure – edges in blue bold arrows to be retained to generate the summary for each topic segment.**

*start* node and the *end* node to ensure defined start and end points of the merged structure. The words from the utterances are iteratively added onto the graph. The words that have the same word form and POS tag are assigned to the same nodes.

**Ambiguity resolver.** Suppose that a new word  $w_i$  that has  $k$  ambiguous nodes where it can be mapped to. The  $k$  ambiguous nodes are referred to as mappable nodes. For every ambiguous mapping candidate, we first find the words to the left and right of the mappable node of the sentences, and then compute the number of words in both the directions that are common to the words in either direction of the word  $w_i$ . Finally,  $w_i$  is mapped to the node that has the highest directed context.

**ILP formulation.** Figure 1 shows the sub-graph (marked using blue bold arrows) that we wish to retain from the merged graph structure to generate a one-sentence summary from several merged utterances. All the sentences generated from each meeting transcript are concatenated to produce the final abstractive summary. We need to maximize the information content of the generated sentence, keeping it grammatical. We model the problem as an integer linear programming (ILP) formulation, similar to the dependency graph fusion as proposed by Filipova and Strube [1]. The directed edges in the graph (binary variables) are represented as  $x_{g,d,l}$ , where  $g$ ,  $d$  and  $l$  denote the governor node, dependent node and the label of an edge, respectively. We maximize the following objective function:

$$\sum_x x_{g,d,l} \cdot p(l | g) \cdot I(d) \cdot \frac{p_x}{N} \quad (1)$$

As shown in Equation (1), we introduce three different terms:  $p(l | g)$ ,  $I(d)$  and  $\frac{p_x}{N}$ . Each relation in a dependency graph consists of the governing node, the dependent node and the relation type. The term  $p(l | g)$  denotes the probabilities of the labels given a governor node,  $g$ . For every node (word and POS) in the entire corpus, the probabilities are represented as the ratio of the sum of the frequency of a particular label and the sum of the frequencies of all the labels emerging from a node. In this work, we calculate these values using Reuters corpora [5] to obtain dominant relations from non-conversational style of text. For example, Table 2 shows the probabilities of outgoing edges from a node, (*produced/VBN*). This term assigns the importance of grammatical relations to a node and only the relations that are more dominant from a node will be preferred. The term  $I(d)$  denotes the informativeness of a node calculated using Hori and Furui’s formula [2]. The last term in Equation (1) is based on the idea of lexical cohesion. Towards the end of any segment, generally, more important discussions might happen that will conclude a particular topic and then start another. In order to take this fact into account, we introduce the term  $\frac{p_x}{N}$ ,

**Table 2: Probabilities of relations from “*produced/VBN*.”**

auxpass	nsubjpass	aux	prep_with	agent	prep_in	advmod
0.286	0.214	0.214	0.071	0.071	0.071	0.071

**Table 3: Content selection evaluation. ROUGE scores (R-2 and R-SU4) and log likelihood score (LL) from the Stanford dependency parser.**

Method	R-2	R-SU4	LL
Our abstractive model	<b>0.048</b>	<b>0.087</b>	<b>-125.73</b>
Our abstractive model (without anaphora resolution)	0.036	0.071	-130.32
Extractive Model (baseline)	0.026	0.044	-136.22

where  $N$  and  $p_x$  denote the total number of extracted utterances in a segment and the position of the utterance (the edge  $x$  belongs to) in the set of  $N$  utterances, respectively.

In order to solve the above ILP problem, we impose a number of constraints. Some of the constraints have been directly adapted from the original ILP formulation [1]. For example, we use the same constraints for restricting one incoming edge per node, as well as we impose the connectivity constraint to ensure a connected graph structure. Further, we restrict the subtree to have just one start edge and one end edge. This helps in preserving one ROOT node, as well as it limits to one *end* node for the generated subtree. We also limit the generated subtree to have a maximum of 15 nodes that controls the length of the summary sentence. We also add few linguistic constraints that ensure the coherence of the output such as every node can have maximum of one determinant, etc. We also impose constraints to prevent cycles in the graph structure, otherwise finding the best path from *start* and *end* nodes might be difficult. The final graph is linearized to obtain a coherent sentence. In the linearization process, we order the nodes based on their original ordering in the utterance.

### 3. EXPERIMENTAL RESULTS

The AMI Meeting corpus contains 20 meeting transcripts in the test set along with their corresponding abstractive (human-written) summaries as well as the annotations of topic segments. ROUGE is used to compare content selection of several approaches. We compared the content selection of our approach to an extractive summarizer [3], which works as a baseline. We also compared our model without using anaphora resolution to see the impact of resolving pronouns. All the summaries were compared against the human-written summaries as reference. The results in Table 3 show that our method outperforms the other techniques on both ROUGE-2 (R2) and ROUGE-SU4 (R-SU4) recall scores. Moreover, we computed a coarse estimate of grammaticality using the log-likelihood score (LL) from the parser. Our technique significantly outperforms the extractive method. In future work, we plan to design an end-to-end framework for summary generation from meetings.

### 4. REFERENCES

- [1] K. Filipova and M. Strube. Sentence Fusion via Dependency Graph Compression. In *Proc. of EMNLP*, pages 177–185, 2008.
- [2] C. Hori and S. Furui. A New Approach to Automatic Speech Summarization. *IEEE Transactions on Multimedia*, 5(3):368–378, 2003.
- [3] G. Murray and G. Carenini. Summarizing Spoken and Written Conversations. In *Proc. of EMNLP*, pages 773–782, 2008.
- [4] G. Murray, G. Carenini, and R. Ng. Generating and Validating Abstracts of Meeting Conversations: a User Study. In *Proc. of INLG*, pages 105–113, 2010.
- [5] T. Rose, M. Stevenson, and M. Whitehead. The Reuters Corpus Volume 1—from Yesterday’s News to Tomorrow’s Language Resources. In *Proc. of LREC*, pages 827–832, 2002.