Abstractive Meeting Summarization with Entailment and Fusion

Yashar Mehdad[*] **Giuseppe Carenini**[*] **Frank W. Tompa**[**] **Raymond T. NG**[*]
Department of Computer Science
[*]University of British Columbia [**]University of Waterloo
{mehdad, carenini, rng}@cs.ubc.ca fwtompa@cs.uwaterloo.ca

Abstract

We propose a novel end-to-end framework for abstractive meeting summarization. We cluster sentences in the input into communities and build an entailment graph over the sentence communities to identify and select the most relevant sentences. We then aggregate those selected sentences by means of a word graph model. We exploit a ranking strategy to select the best path in the word graph as an abstract sentence. Despite not relying on the syntactic structure, our approach significantly outperforms previous models for meeting summarization in terms of informativeness. Moreover, the longer sentences generated by our method are competitive with shorter sentences generated by the previous word graph model in terms of grammaticality.

1 Introduction

The huge amount of data generated every day in meetings calls for developing automated methods to efficiently process these data to meet users' needs. Automatic summarization is a popular task that can help users to browse a large amount of recorder speech in text format. This paper tackles the task of recorded meeting summarization, addressing the key limitations of existing approaches by proposing the following contributions:

1) Various approaches that were recently developed for meeting summarization (such as (Gillick et al., 2009; Garg et al., 2009)) focus on extracting important sentences (or dialogue acts) from speech transcripts, either manual transcripts or automatic speech recognition (ASR) output. However, it has been observed in the context of meeting summarization users generally prefer concise abstracts over extracts, and abstracts lead to higher objective task scores; likewise automatic abstractive summaries are preferred in comparison with human extracts (Murray et al., 2010). Moreover, most of the abstractive summarization approaches focus on one component of the system, such as generation (e.g., (Genest and Lapalme, 2010)) or content selection (e.g., (Murray et al., 2012)), instead of developing the full framework for abstractive summarization. To address these limitations, as the main contribution of this paper, we propose a full pipeline to generate an abstractive summary for each meeting transcript. Our system is similar to that of Murray et al. (2010) in terms of generating abstractive summaries for meeting transcripts. However, we take a lighter supervision for the content selection phase and a different approach towards the language generation phase, which does not rely on the conventional Natural Language Generation (NLG) architecture (Reiter and Dale, 2000).

2) We propose a word graph based approach to aggregate and generate the abstractive sentence summary. Our work extends the word graph method proposed by Filippova (2010) with the following novel contributions: *i)* We take advantage of lexical knowledge to merge similar nodes by finding their relations in WordNet; *ii)* We generate new sentences through generalization and aggregation of the original ones, which means that our generated sentences are not necessarily composed of the original words; and *iii)* We adopt a new ranking strategy to select the best path in the graph by taking the information content and the grammaticality (i.e. fluency) of the sentence into consideration.

3) In order to generate an abstract summary for a meeting, we have to be able to capture the overall content of the conversation. This can be done by identifying the essential content from the most informative sentences using the semantic relations among all sentences in a meeting transcript. How-

Proceedings of the 14th European Workshop on Natural Language Generation, pages 136–146,
Sofia, Bulgaria, August 8-9 2013. ©2013 Association for Computational Linguistics

ever, most current methods disregard such relations in favor of statistical models of word distributions and frequencies. Moreover, the data from meeting transcripts often contains many highly redundant sentences. As one of the key contributions of this paper, we propose to build a multi-directional entailment graph over the sentences to identify and select relevant information from the most informative sentences.

4) The textual data from meeting conversation transcripts are typically in a casual style and do not exhibit a clear syntactic structure with proper grammar and spelling. Therefore, abstractive summarization approaches developed for formal texts, such as scientific or news articles, often are not satisfactory when dealing with informal texts. Our proposed method for abstractive meeting summarization requires minimal syntactic and structural information and is robust in dealing with text that suffers from transcription errors, ill-formed sentences and unknown lexical choices such as typically formed in meeting transcripts.

We evaluate our system over meeting transcripts. Our result compares favorably to the result of previous extractive and abstractive approaches in terms of information content. Moreover, we show that our method can generate longer sentences with competitive grammaticality scores, in comparison with previous abstractive approaches. Furthermore, we evaluate the impact of each component of our system through an ablation test. As an additional result of our experiments, we also show that using semantic relations (entailment graph) is important in efficiently performing the final step of our summarization pipeline (i.e., the sentence fusion).

2 Abstractive Summarization Framework

Similar to Murray et al. (2010), our goal is to generate a meeting summary, i.e. a set of sentences, that could capture the semantics of the meeting. While (Murray et al., 2010) requires extensive annotations to train several classifiers to detect important sentences, opinions and dialog acts, we only use a subset of that annotation, which includes a human abstract and links from each sentence in the abstract to the source meeting sentences. In addition, instead of generating an abstractive sentence via the conventional NLG pipeline (Reiter and Dale, 2000), we exploit a word graph approach.

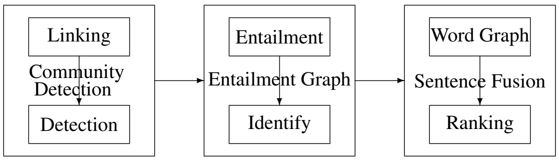

Figure 1: Meeting summarization framework.

As shown in Figure 1, our framework consists of three main components, which we describe in more detail in the following sections.

2.1 Community Detection

While some abstractive summary sentences are very similar to original sentences from the meeting transcript, others can be created by aggregating and merging multiple sentences into an abstract sentence. In order to generate such a sentence, we need to identify which sentences from the original meeting transcript should be combined in generated abstract sentences. This task can be considered as the first step of abstractive meeting summarization and is called “abstractive community detection (ACD)” (Murray et al., 2012). To perform ACD, we follow the same method proposed by Murray et al. (2012), in two steps:

First, we classify sentence pairs according to whether or not they should be realized by a common abstractive sentence. For each pair, we extract its structural and linguistic features, and we train a logistic regression classifier over all our training data (described in Section 3.1) exploiting such features. We run the trained classier over sentence pairs, predicting abstractive links between sentences in the document. The result can be represented as an undirected graph where nodes are the sentences, and edges represent whether two nodes are linked.

Second, we have to identify which nodes (i.e., sentences from the meeting transcript) can be clustered as a community to generate an abstract sentence. For this purpose, we apply the CONGA algorithm (Gregory, 2007) for community detection that uses betweenness to detect communities in a graph. The betweenness score for an edge is the number of shortest paths between pairs of nodes in the graph that run along that edge.

If a sentence is not connected to at least one other sentence in the first step, it’s assigned to its own singleton community.

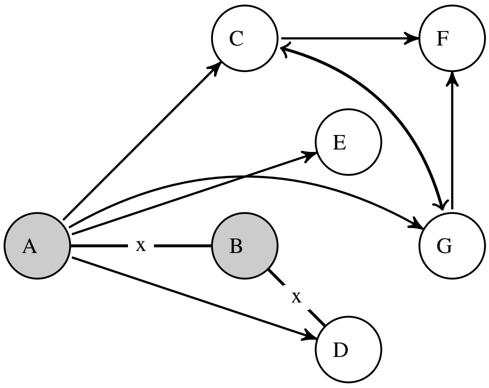

Figure 2: Building an entailment graph over sentences. Arrows and "x" represent the entailment direction and unknown cases respectively.

2.2 Entailment Graph

Sentences in a community often include redundant information which are semantically equivalent but vary in lexical choices. By identifying the semantic relations between the sentences in each community, we can discover the information in one sentence that is semantically equivalent, novel, or more/less informative with respect to the content of the other sentences.

Similar to earlier work (Lloret et al., 2008; Mehdad et al., 2010; Berant et al., 2011; Adler et al., 2012; Mehdad et al., 2013), we set this problem as a variant of the Textual Entailment (TE) recognition task (Dagan and Glickman, 2004). We build an entailment graph for each community of sentences, where nodes are the linked sentences and edges are the entailment relations between nodes. Given two sentences (s_1 and s_2), we aim at identifying the following cases:
i) s_1 and s_2 express the same meaning (*bidirectional* entailment). In such cases one of the sentences should be eliminated;
ii) s_1 is more informative than s_2 (*unidirectional* entailment). In such cases, the entailing sentence should replace or complement the entailed one;
iii) s_1 contains facts that are not present in s_2, and vice-versa (the "*unknown*" cases in TE parlance). In such cases, both sentences should remain.

Figure 2 shows how entailment relations can help in selecting the sentences by removing the redundant and less informative ones. As we show in the figure, the sentence "*A*" entails "*E*", "*F*" and "*G*", but not "*B*". So we can keep "*A*" and "*B*" and eliminate others. For example, the sentence "*we should discuss about the remote control and its color*" entails "*about the remote*", "*let's talk about the remote*" and "*um remote's color*", but not "*remote's size is also important*". So we can keep "*we should discuss about the remote control and its color*" and "*remote's size is also important*" and eliminate the others. In this way, TE-based sentence identification can be designed to distinguish meaning-preserving variations from true divergence, regardless of lexical choices and structures.

Similar to previous approaches in TE (*e.g.*, (Berant et al., 2011)), we use a supervised method. To train and build the entailment graph, we perform three steps described in the following subsections.

2.2.1 Training set collection

In the last few years, TE corpora have been created and distributed in the framework of several evaluation campaigns, including the Recognizing Textual Entailment (RTE) Challenge[1] and Cross-lingual textual entailment for content synchronization[2] (Negri et al., 2012). However, such datasets cannot directly support our application, since the RTE datasets are often composed of longer well-formed sentences and paragraphs (Bentivogli et al., 2009; Negri et al., 2011).

In order to collect a dataset that is more similar to the goal of our entailment framework, we select a subset of the sixth and seventh RTE challenge main task (*i.e.*, RTE within a Corpus). Our dataset choice is based on the following reasons: *i*) the length of sentence pairs in RTE6 and RTE7 is shorter than the others, and *ii*) RTE6 and RTE7 main task datasets are originally created for summarization purpose, which is closer to our work. We sort the RTE6 and RTE7 dataset pairs based on the sentence length and choose the first 2000 samples with an equal number of positive and negative examples. The average length of words in our training data is 7 words. There are certainly some differences between our training set and our sentences from the meeting corpus. However, the collected training samples was the closest available dataset to our needs.

2.2.2 Feature representation and training

Working with meeting transcripts imposes some constraints on feature selection. Meeting transcripts are not often well-formed in terms of sen-

[1] http://pascallin.ecs.soton.ac.uk/Challenges/RTE/
[2] http://www.cs.york.ac.uk/semeval-2013/task8/

tence structure and contain errors. This limits our features to the lexical level. Fortunately, lexical models are less computationally expensive and easier to implement and often deliver a strong performance for RTE (Sammons et al., 2011).

Our entailment decision criterion is based on similarity scores calculated with a sentence-to-sentence matching process. Each example pair of sentences (s_1 and s_2) is represented by a feature vector, where each feature is a specific similarity score estimating whether s_1 entails s_2.

We compute 18 similarity scores for each pair of sentences. Before aggregating the similarity scores to form an entailment score, we normalize the similarity scores by the length of s_2 (in terms of lexical items), when checking the entailment direction from s_1 to s_2. In this way, we can estimate the portion of information/facts in s_2 which is covered by s_1.

The first five scores are computed based on the exact lexical overlap between the phrases: word overlap, edit distance, ngram-overlap, longest common subsequence and Lesk (Lesk, 1986). The other scores were computed using lexical resources: WordNet (Fellbaum, 1998), VerbOcean (Chklovski and Pantel, 2004), paraphrases (Denkowski and Lavie, 2010) and phrase matching (Mehdad et al., 2011). We used WordNet to compute the word similarity as the least common subsumer between two words considering the synonymy-antonymy, hypernymy-hyponymy, and meronymy relations. Then, we calculated the sentence similarity as the sum of the similarity scores of the word pairs in Text and Hypothesis, normalized by the number of words in Hypothesis. We also use phrase matching features described in (Mehdad et al., 2011) which consists of phrasal matching at the level on ngrams (1 to 5 tokens). The rationale behind using different entailment features is that combining various scores will yield a better model (Berant et al., 2011).

To combine the entailment scores and optimize their relative weights, we train a Support Vector Machine binary classifier, SVMlight (Joachims, 1999), over an equal number of positive and negative examples. This results in an entailment model with 95% accuracy over 2-fold and 5-fold cross-validation, which further proves the effectiveness of our feature set for this lexical entailment model. The reason that we gained a very high accuracy is because our selected sentences are a subset of RTE6 and RTE7 with a shorter length (fewer words) which makes the entailment recognition task much easier than recognizing entailment between paragraphs or long sentences.

2.2.3 Entailment graph edge labeling

Since our training examples are labeled with binary judgments, we are not able to train a three-way classifier. Therefore, we set the edge labeling problem as a two-way classification task that casts multidirectional entailment as a unidirectional problem, where each pair is analyzed checking for entailment in both directions (Mehdad et al., 2012). In this condition, each original test example is correctly classified if both pairs originated from it are correctly judged ("*YES-YES*" for bidirectional,"*YES-NO*" and "*NO-YES*" for unidirectional entailment and "*NO-NO*" for unknown cases). Two-way classification represents an intuitive solution to capture multidimensional entailment relations.

2.2.4 Identification and selection

By assigning all entailment relations between the extracted sentence pairs, we identify relevant sentences to eliminate the redundant (in terms of meaning) and less informative ones. In order to perform this task we follow a set of rules based on the graph edge labels. Note that since entailment is a transitive relation, our entailment graph is transitive *i.e.*, if entail(s_1,s_2) and entail(s_2,s_3) then entail(s_1,s_3) (Berant et al., 2011).

Rule 1) Among the nodes that are connected with bidirectional entailment (semantically equivalent nodes) we keep only the one with more outgoing bidirectional and unidirectional entailment relations;

Rule 2) If there is a chain of entailing nodes, we keep the one that is the root of the chain and eliminate others.

2.3 Multi-sentence Fusion

Sentence fusion is a well-known challenge for summarization systems. In this phase, our goal is to generate an understandable informative sentence that maximally captures the content of the sentences in a sentence community.

There are several ways of generating an abstract sentence (e.g. (Barzilay and McKeown, 2005; Liu and Liu, 2009; Ganesan et al., 2010; Murray et al., 2010)); however, most of them rely heavily on the syntactic structure. We believe that such

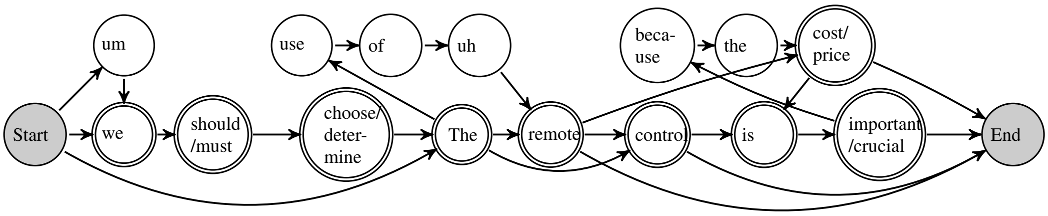

Figure 3: Word graph constructed from sentences (1-4) and possible fusion paths. Double line nodes represent merged words in the graph.

approaches are suboptimal, especially in dealing with written conversational data (e.g., email) or the data from speech transcripts, whether manual transcription or automatic speech recognition output. Instead, we apply an approach that does not rely on syntax, nor on a standard NLG architecture. More specifically, we build a word graph from all the words of the sentences in a community and aggregate them in order to generate a new abstractive sentence.

We perform the task of multi-sentence fusion in two steps. First, we construct a word graph over sentences in each community that were selected from the entailment graph. Second, we rank the valid paths in the word graph and select the top path as the abstract sentence summary.

2.3.1 Constructing a Word Graph

In order to construct a word graph, our model extends the word graph method proposed by Filippova (2010) with the following novel contributions:

1- The basic word graph method disregards semantic and lexical relations between the words in constructing the word graph, in favor of redundancy and word frequencies. To move beyond such limitation, we take advantage of lexical knowledge to map the similar nodes by finding their relations in WordNet. In this way, for example, two synonym words can be mapped into the same node.

2- Filippova's approach is essentially extractive in nature, which means the generated sentence is composed by the same words from the original sentences. We move beyond this by generating new sentences through generalization and aggregation of the original ones. This means that our generated sentences are not necessarily composed of the original words. In this way, we are one step closer to abstractive summarization.

3- Our proposed method aggregates and generates new readable sentences, regardless of their lengths, that can semantically imply several original sentences, by taking the information content and the readability (i.e. fluency) of the sentence into consideration.

Following Filippova's method, given a set of related sentences, we build a word graph by iteratively adding sentences to it. Figure 1 illustrates a small graph composed of 4 sentences, including the start and end nodes.

1- *we must determine the use of uh remote.*
2- *The remote control is important because the cost.*
3- *um we should choose the control.*
4- *The remote price is crucial.*

As one of the main steps of word graph construction, we merge the words that have the same POS tags under the following conditions:

1) The words are identical (e.g. *"remote"*).

2) The words are synonyms. The replacement choice is based on the word's commonality, i.e. *tfidf* (e.g. *"important"* and *"crucial"*).

3) The words form a hypernym/hyponym pair or share a common hypernym. Both words are replaced by the hypernym (e.g. *"price"* and *"cost"*).

4) The words are in an entailment relation. Both words are replaced by the entailed one (e.g. *"pay"* and *"buy"*).

Note that, similar to Filippova's approach, where merging is ambiguous we check the context (a word before and after in the sentence and the neighboring nodes in the graph) and select the candidate that has larger overlap in the context, or the one with a greater frequency. Similar to the original word graph model, we connect adjacent words with directed edges. For the new nodes or unconnected nodes, we draw an edge with a weight of 1. In contrast, weights between already connected nodes are increased by 1.

2.3.2 Path Selection and Ranking

The word graph, as described above, will generate many sequences connecting start and end. However, it is likely that most of the paths are not readable. Since we are aiming at generating a good abstractive sentence, some constraints need to be considered.

A good abstractive sentence should cover most of the concepts that exist in the original sentences. Moreover, it should be grammatically correct.

In order to satisfy these constraints we adopt a ranking strategy that combines the characteristics of a good summary sentence. To filter ungrammatical sentences, we prune the paths in which a verb does not exist. Our ranking formulation is summarized as below:

Fluency: Our word graph process generates many possible paths as abstractive summaries. We need now to decide which of these paths are more readable and fluent. As in other areas of NLP (e.g. machine translation and speech recognition), the answer can be estimated by a language model. We assign a probability $Pr(P)$ to each path P based on a n-gram language model.

$$Pr(P) = \prod_{i=1}^{m} Pr(p_i|p_1^{i-1}) \approx \prod_{i=1}^{m} Pr(p_i|p_{i-n+1}^{i-1})$$
$$\approx \sum_{i=1}^{m} -logPr(p_i|p_{i-n+1}^{i-1})$$

Coverage: To identify the summary with the highest coverage, we propose a second score that estimates the number of nouns that appear in the path. In order to reward the ranking score to cover more salient nouns, we also consider the *tfidf* score of nouns in the coverage formulation.

$$Coverage(P) = \frac{\sum_{p_i \in P} \textit{tfidf}(p_i)}{\sum_{p_i \in G} \textit{tfidf}(p_i)}$$

where the p_i are nouns and G is the graph.

Edge weight: As a third score, we adopt the Filippova's edge weighting formulation $w(p_i, p_j)$ and define the edge weight of the path $W(P)$ as below:

$$w(p_i, p_j) = \frac{\textit{freq}(p_i) + \textit{freq}(p_j)}{\sum_{\substack{P \in G \\ p_i, p_j \in P}} \textit{diff}(P, p_i, p_j)^{-1}}$$

$$W(P) = \frac{\sum_{i=1}^{m-1} w(p_i, p_{i+1})}{m-1}$$

where the function $\textit{diff}(P, p_i, p_j)$ refers to the distance between the offset positions $pos(P, p_i)$ of words p_i and p_j in path P and is defined as $|pos(P, p_j) - pos(P, p_i)|$ and m is the number of words in path P.

Ranking score: In order to generate a summary sentence that combines the scores above, we employ a ranking model. The purpose of such a model is three-fold: i) to generate a more readable and grammatical sentence; ii) to cover the content of original sentences optimally; and iii) to favor strong connections between the concepts. Therefore, the final ranking score of path P is calculated over the normalized scores as:

$$Score(P) = \frac{Pr(P) \times Coverage(P)}{W(P)}$$

We select all the paths that contain at least one verb and rerank them using our proposed ranking function to find the best path as the summary of the original sentences.

3 Experiments and Results

We now describe a preliminary, formative evaluation of our framework, whose main goal is to assess strengths and weaknesses of the various components and generate ideas for further developments.

3.1 Dataset

To verify the effectiveness of our approach, we experiment with the AMI meeting corpus (Carletta et al., 2005) that consists of 140 multi-party meetings with a wide range of annotations, including abstactive summaries for each meeting and links from each sentence in the summary to the set of sentences in the original transcripts that sentence is summarizing. We use this information as our gold standard since it provides many examples in which a set of sentences in the meeting (a community) is linked to a human written sentence summarizing that community.

In our experiments, we use human authored transcripts. Note that our approach is not specific to conversations, however it is designed to deal with ill-formed sentences and structural errors. Moreover, the first two components of our system are supervised, while the word graph model is completely unsupervised and domain independent.

In order to train our logistic regression classifier for the first phase of our pipeline, we randomly select a training set that consists of 98 meetings. Note that there are about one million sentence pair instances in the training set since we consider every pairing of sentences within a meeting. The rest is selected as a test set for the evaluation phase.

3.2 Experimental Settings

For preprocessing our dataset we use OpenNLP[3] for tokenization and part-of-speech tagging. When the number of sentences in each community is more than 10 (which happens in around 10% of the cases), the community is first clustered using the MajorClust (Stein and Niggemann, 1999) algorithm when sentences are represented as normalized *tfidf* vectors and the similarity between the sentences is measured using cosine similarity. Then, each cluster is treated as a separate community. The clustering guarantees a higher overlap between the sentences as well as a word graph with fewer paths. Next, we construct a word graph over each cluster and rank the valid paths. We choose the first ranked path as the abstractive summary of the cluster. For our language model, we use a tri-gram smoothed language model trained using the newswire text provided in the English Gigaword corpus (Graff and Cieri,).

3.3 Evaluation Metrics

To evaluate performance, we use the ROUGE-1 and ROUGE-2 (unigram and bigram overlap) F1 score, which correlate well with human rankings of summary quality (Lin and Hovy, 2003). We also ignore stopwords to reduce the impact of high overlap when matching them.

Furthermore, to evaluate the grammaticality of our generated summaries in comparison with the original word graph method, following common practice (Barzilay and McKeown, 2005), we randomly selected 10 meeting summaries (total 150 sentences). Then, we asked annotators to give one of three possible ratings for each sentence in a summary based on grammaticality: perfect (2 pts), only one mistake (1 pt) and not acceptable (0 pts), ignoring the capitalization or punctuation. Each meeting was rated by two annotators (Computer Science graduate students).

Models	ROUGE-1	ROUGE-2
MMR-centroid	18	3
MMR-cosine	21	-
ILP	24	-
TextRank	25.0	4.4
ClusterRank	27.5	**5.1**
Orig. word graph	26.9	3.8
Our model (-ent)	**32.3**	**4.8**
Our model (GC)	**32.1**	4.0
Our model (full)	**28.7**	4.2

Table 1: Performance of different summarization algorithms on human transcripts for meeting conversations. [5]

3.4 Baselines

We compare our approach with various extractive baselines: 1) MMR-centroid system (Carbonell and Goldstein, 1998); 2) MMR-cosine system (Gillick et al., 2009); 3) ILP-based system (Gillick et al., 2009); 4) TextRank system (Mihalcea and Tarau, 2004); and 5) ClusterRank system (Garg et al., 2009) and with one abstractive baseline: 6) Original word graph model (Orig. word graph) (Filippova, 2010).

In order to measure the effectiveness of different components, we also evaluated our system using human-annotated sentence communities (GC) in comparison with our community detection model (full). Moreover, we measure the performance of our system (GC) ablating the entailment module (-ent).

3.5 Results

Table 1 shows the results for our proposed approach in comparison with these strong baselines for meeting summarization. The results show that our model outperforms the baselines significantly[4] for ROUGE-1 over human transcripts for meeting conversations, which proves the effectiveness of our approach in dealing with summarization of

[3] http://opennlp.apache.org/

[5] The MMR-cosine and ILP systems did not report the ROUGE-2 score.

[4] The statistical significance tests was calculated by approximate randomization described in (Yeh, 2000).

Models	Read.	R=2	R=1	R=0	Avg Len.
Orig. word graph	1.41	55%	32%	13%	8
Our model	1.34	47%	39%	14%	14

Table 2: Average rating and distribution over rating scores for abstractive word graph models.

meeting conversations. However, the ClusterRank and TextRank systems outperform our model for ROUGE-2 score. This can be due to word merging and word replacement choices in the word graph construction (see Section 2.3.1), which sometimes changes a word in a bigram and consequently decreases the bigram overlap score. A more detailed analysis of this problem is left as future work.

Note that there is a drop in ROUGE score when we use entailment in our system in comparison with ablating the entailment phase (-ent). This is mainly due to the fact that the entailment phase filters equivalent sentences. This affects the results negatively when such filtered sentences share many common words with our human-authored abstracts. We believe that this drop is partly associated with our evaluation metric rather than meaning. In other words, we expect no difference in performance when a human evaluation is applied. However, the entailment phase helps in improving the efficiency of our pipeline significantly. If each graph has e edges, n nodes, and p paths, then finding all the paths results in time complexity $O((np + 1)(e + n))$, using depth-first search. Decreasing the number of sentences will reduce the number of nodes and edges, which leads to the smaller number of paths. This is even more significant when there are many sentences in a community in comparison with the gold standard. Note that it's impossible to finish the graph building phase after 12 hours on a 2.3 GHz quad-core machine without performing the entailment phase, when we use our community detection model. This would be especially problematic in a real-time setting.

Comparing the gold standard sentence communities (GC) and our fully automatic system, we can notice that inaccuracies in the community detection phase affects the overall performance. However, using our community detection model, we still outperform the previous models significantly.

Table 2 shows grammaticality scores, distributions over the three scores and average sentence lengths in tokens. The results demonstrate that 47% of the sentences generated by our method are grammatically correct and 39% of the generated sentences are almost correct. In comparison with the original word graph method, our model reports slightly lower results for the grammaticality score and the percentage of correct sentences. However, considering the correlation between sentence length and grammatical complexity, our model is capable of generating longer sentences with more information content (according to ROUGE) and competitive grammaticality scores.

4 Discussion

After analyzing the results and through manual verification of some cases, we observe that our approach produces some interestingly successful examples. Nevertheless, it appears that the performance is still far from satisfactory. This leaves an interesting challenge for the research community to tackle. We have identified five different sources of error:

Type 1: Abstractive human-authored summaries: the nature of our method is based on extracting the relevant sentences and generating an abstract sentence by aggregating such sentences. Also due to this, our generated abstracts are often informal and closer to the transcripts' style. However, in many cases, the human-written summaries are composed by understanding the original sentences and produce a formal style abstract sentence, often using a different vocabulary and structure. For example:

> Human-authored: *The industrial designer and user interface designer presented the prototype they created, which was designed to look like a banana.*
> System: *Working on the principle of a fruit it's basically designed around a banana.*

Type 2: Evaluation method: The current evaluation methods fail to capture the meaning and relies only on matching the words at uni- or bigram level. Therefore, we believe that a manual evaluation can reveal more potential of our system in generating abstractive summaries that are closer to human-written summaries.

Human-authored: *the project manager recapped the decisions made in the previous meeting.*
System: *I told you guys about the three new requirements ... so that was the last meeting.*

Type 3: Subjective abstractive summaries: often it is not easy for humans to agree on one summary for a meeting. It is well known that inter-annotator agreement is quite low for the summarization task (Mani, 2001). For example:

Human-authored 1: *They do tool training with a whiteboard and each person introduces themselves and draws their favorite animal on the board.*
Human-authored 2: *The group introduced themselves to each other and acquainted themselves with the meeting-room materials by drawing on the whiteboard.*
System: *We are gonna know each other and then draw your little animal.*

Type 4: Speaker information: since the nature of our method is based on extracting the relevant sentences or speaker utterances, we do not take the speaker information into consideration. However, the human-written summaries for meetings take the speaker into account. We plan to extend our framework to include this feature. For example:

Human-authored: *The project manager opened the meeting and stated the agenda to the team members.*
System: *I hope you're ready for this functional design meeting know at the end projects requirement.*

Type 5: Transcription errors: as mentioned before, the meeting transcripts often contain structure, grammar, vocabulary choice and dictation errors. This always raises more challenges for algorithms dealing with such texts. For example:

Transcript: *if it i if it isn't more expensive for us to k make because as far as I understand it.*

In light of this analysis, we conclude that a more comprehensive evaluation method (e.g., human evaluation), including speaker information in the pipeline and using text normalization techniques to reduce the effects of noisy transcripts can better reveal the potential of our system in dealing with meeting summarization.

5 Conclusion and Future Work

In this paper, we study the problem of abstractive meeting summarization, and propose a novel framework to generate summaries composed of grammatical sentences. Within this framework, this paper makes three main contributions. First, in contrast with most current methods based on fully extractive models, we propose to take advantage of a word graph model for sentence fusion to generate abstractive summary sentences. Second, beyond most of the current approaches which disregard semantic information, we integrate semantics by means of building textual entailment graphs over sentence communities. Third, our framework uses minimal syntactic information in comparison with previous methods and does not require a domain specific, engineered conventional NLP component.

We successfully applied our framework over a challenging meeting dataset, the AMI corpus. Some significant improvements over our dataset, in comparison with previous methods, demonstrates the potential of our approach in dealing with meeting summarization. Moreover, we prove that our model can generate longer sentences with only a minimal loss in grammaticality.

In light of the results of our preliminary formative evaluation, future work will address the improvement of the community detection and sentence fusion phases. On the one hand, we plan to improve our community detection graph by adding more relevant features into our current supervised model. On the other hand, we plan to incorporate a better source of lexical knowledge in the word graph construction (e.g., YAGO or DBpedia). We are also interested in improving our ranking model by assigning tuned weights to each component. In addition, we are exploring the replacement of pronouns by their referents (e.g., replacing *"I"* by the name or role of the speaker) to improve both the entailment and word graph models. Once we will have explored all these improvements, we plan to run more comprehensive human evaluations.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments and suggestions to improve the paper, our annotators for their valuable work, and the NSERC Business Intelligence Network for financial support.

References

Meni Adler, Jonathan Berant, and Ido Dagan. 2012. Entailment-based text exploration with application to the health-care domain. In *Proceedings of the ACL 2012 System Demonstrations*, ACL '12, pages 79–84, Stroudsburg, PA, USA. Association for Computational Linguistics.

Regina Barzilay and Kathleen R. McKeown. 2005. Sentence Fusion for Multidocument News Summarization. *Comput. Linguist.*, 31(3):297–328, September.

Luisa Bentivogli, Ido Dagan, Hoa Trang Dang, Danilo Giampiccolo, and Bernardo Magnini. 2009. The Fifth PASCAL Recognizing Textual Entailment Challenge. In *Proc Text Analysis Conference (TAC09.*

Jonathan Berant, Ido Dagan, and Jacob Goldberger. 2011. Global Learning of Typed Entailment Rules. In *Proceedings of ACL*, Portland, OR.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '98, pages 335–336, New York, NY, USA. ACM.

Jean Carletta, Simone Ashby, Sebastien Bourban, Mike Flynn, Thomas Hain, Jaroslav Kadlec, Vasilis Karaiskos, Wessel Kraaij, Melissa Kronenthal, Guillaume Lathoud, Mike Lincoln, Agnes Lisowska, and Mccowan Wilfried Post Dennis Reidsma. 2005. The AMI meeting corpus: A pre-announcement. In *Proc. MLMI*, pages 28–39.

Timothy Chklovski and Patrick Pantel. 2004. VerbOcean: Mining the Web for Fine-Grained Semantic Verb Relations. In Dekang Lin and Dekai Wu, editors, *Proceedings of EMNLP 2004*, pages 33–40, Barcelona, Spain, July. Association for Computational Linguistics.

I. Dagan and O. Glickman. 2004. Probabilistic Textual Entailment: Generic applied modeling of language variability. In *PASCAL Workshop on Learning Methods for Text Understanding and Mining.*

Michael Denkowski and Alon Lavie. 2010. METEOR-NEXT and the METEOR paraphrase tables: improved evaluation support for five target language. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR*, WMT '10, pages 339–342, Stroudsburg, PA, USA. Association for Computational Linguistics.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Katja Filippova. 2010. Multi-sentence compression: finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 322–330, Stroudsburg, PA, USA. Association for Computational Linguistics.

Kavita Ganesan, ChengXiang Zhai, and Jiawei Han. 2010. Opinosis: a graph-based approach to abstractive summarization of highly redundant opinions. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 340–348, Stroudsburg, PA, USA. Association for Computational Linguistics.

Nikhil Garg, Benoit Favre, Korbinian Reidhammer, and Dilek Hakkani Tür. 2009. ClusterRank: A Graph Based Method for Meeting Summarization. Idiap-RR Idiap-RR-09-2009, Idiap, P.O. Box 592, CH-1920 Martigny, Switzerland, 6.

Pierre-Etienne Genest and Guy Lapalme. 2010. Text Generation for Abstractive Summarization. In *Proceedings of the Third Text Analysis Conference*, Gaithersburg, Maryland, USA. National Institute of Standards and Technology, National Institute of Standards and Technology.

Dan Gillick, Korbinian Riedhammer, Benoit Favre, and Dilek Hakkani-tr. 2009. A global optimization framework for meeting summarization. In *Proc. IEEE ICASSP*, pages 4769–4772.

David Graff and Christopher Cieri. English Gigaword Corpus—, year = 2003, institution = Linguistic Data Consortium, address = Philadelphia,. Technical report.

Steve Gregory. 2007. An Algorithm to Find Overlapping Community Structure in Networks. In *Proceedings of the 11th European conference on Principles and Practice of Knowledge Discovery in Databases*, PKDD 2007, pages 91–102, Berlin, Heidelberg. Springer-Verlag.

T. Joachims. 1999. Making large-Scale SVM Learning Practical. LS8-Report 24, Universität Dortmund, LS VIII-Report.

Michael Lesk. 1986. Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone. In *Proceedings of the 5th annual international conference on Systems documentation*, SIGDOC '86, pages 24–26, New York, NY, USA. ACM.

Chin-Yew Lin and Eduard Hovy. 2003. Automatic evaluation of summaries using N-gram cooccurrence statistics. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology - Volume 1*, NAACL '03, pages 71–78, Stroudsburg, PA, USA. Association for Computational Linguistics.

Fei Liu and Yang Liu. 2009. From extractive to abstractive meeting summaries: can it be done by sentence compression? In *Proceedings of the ACL-IJCNLP 2009 Conference Short Papers*, ACLShort '09, pages 261–264, Stroudsburg, PA, USA. Association for Computational Linguistics.

Elena Lloret, Óscar Ferrández, Rafael Muñoz, and Manuel Palomar. 2008. A Text Summarization Approach under the Influence of Textual Entailment. In *NLPCS*, pages 22–31.

I. Mani. 2001. *Automatic summarization*. Natural Language Processing, 3. J. Benjamins Publishing Company.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2010. Towards cross-lingual textual entailment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, HLT '10, pages 321–324, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2011. Using bilingual parallel corpora for cross-lingual textual entailment. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1336–1345, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yashar Mehdad, Matteo Negri, and Marcello Federico. 2012. Detecting semantic equivalence and information disparity in cross-lingual documents. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers - Volume 2*, ACL '12, pages 120–124, Stroudsburg, PA, USA. Association for Computational Linguistics.

Yashar Mehdad, Giuseppe Carenini, and Raymond NG T. 2013. Towards Topic Labeling with Phrase Entailment and Aggregation. In *Proceedings of NAACL 2013*, pages 179–189, Atlanta, USA, June. Association for Computational Linguistics.

R. Mihalcea and P. Tarau. 2004. TextRank: Bringing order into texts. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, July.

Gabriel Murray, Giuseppe Carenini, and Raymond Ng. 2010. Generating and validating abstracts of meeting conversations: a user study. In *Proceedings of the 6th International Natural Language Generation Conference*, INLG '10, pages 105–113, Stroudsburg, PA, USA. Association for Computational Linguistics.

Gabriel Murray, Giuseppe Carenini, and Raymond Ng. 2012. Using the omega index for evaluating abstractive community detection. In *Proceedings of Workshop on Evaluation Metrics and System Comparison for Automatic Summarization*, pages 10–18, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matteo Negri, Luisa Bentivogli, Yashar Mehdad, Danilo Giampiccolo, and Alessandro Marchetti. 2011. Divide and conquer: crowdsourcing the creation of cross-lingual textual entailment corpora. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, EMNLP '11, pages 670–679, Stroudsburg, PA, USA. Association for Computational Linguistics.

Matteo Negri, Alessandro Marchetti, Yashar Mehdad, Luisa Bentivogli, and Danilo Giampiccolo. 2012. Semeval-2012 task 8: cross-lingual textual entailment for content synchronization. In *Proceedings of the First Joint Conference on Lexical and Computational Semantics - Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*, SemEval '12, pages 399–407, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. 2000. *Building natural language generation systems*. Cambridge University Press, New York, NY, USA.

Mark Sammons, V.G.Vinod Vydiswaran, and Dan Roth. 2011. Recognizing textual entailment. In *Multilingual Natural Language Applications: From Theory to Practice*. Prentice Hall, Jun.

Benno Stein and Oliver Niggemann. 1999. On the Nature of Structure and Its Identification. In *Proceedings of the 25th International Workshop on Graph-Theoretic Concepts in Computer Science*, WG '99, pages 122–134, London, UK, UK. Springer-Verlag.

Alexander Yeh. 2000. More accurate tests for the statistical significance of result differences. In *Proceedings of the 18th Conference on Computational Linguistics - Volume 2*, COLING '00, pages 947–953. Association for Computational Linguistics.