Unsupervised Abstractive Meeting Summarization with Multi-Sentence Compression and Budgeted Submodular Maximization

Guokan Shang[1,2], Wensi Ding[1], Zekun Zhang[1], Antoine J.-P. Tixier[1], Polykarpos Meladianos[1,3], Michalis Vazirgiannis[1,3], Jean-Pierre Lorré[2]

[1]DaSciM Team, Computer Science Laboratory (LIX), École Polytechnique, Palaiseau, France
[2]LINAGORA, Paris La Défense, France
[3]Athens University of Economics and Business, Athens, Greece

Introduction

Spontaneous multi-party meeting speech transcription is made of often ill-formed and ungrammatical text fragments (*utterances*) ⇒ summarizing requires approaches that differ from traditional document summarization.

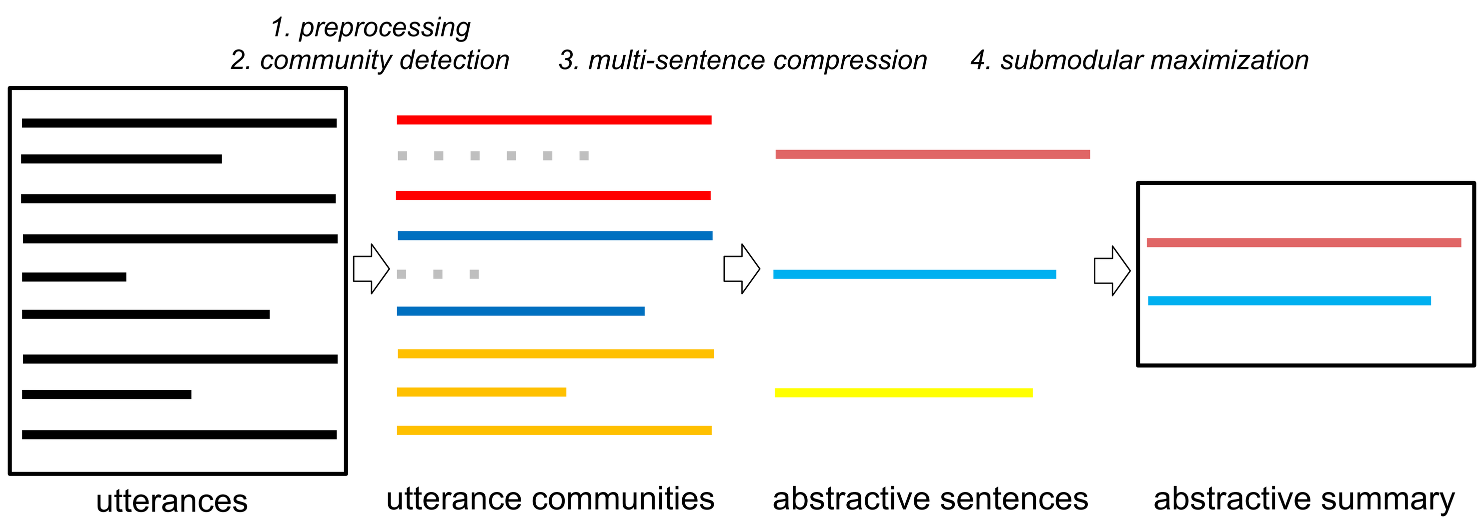


Figure 1: Overarching system pipeline.

1. Preprocessing & 2. Community Detection

▶ Filler words are discarded. *uh-huh, okay well, by the way*
▶ Consecutive stopwords at the head and tail are stripped.
→ Utterances containing less than 3 non-stopwords are pruned out.

Goal: group together the utterances that should be summarized by a common abstractive sentence [Murray et al. 2012].
▶ Utterances → TFIDF → LSA → k-means → communities

Word Scoring with Graph Degeneracy

Keywords are influential spreaders within their word co-occurrence network, better identified by CoreRank score [Tixier et al. 2016].

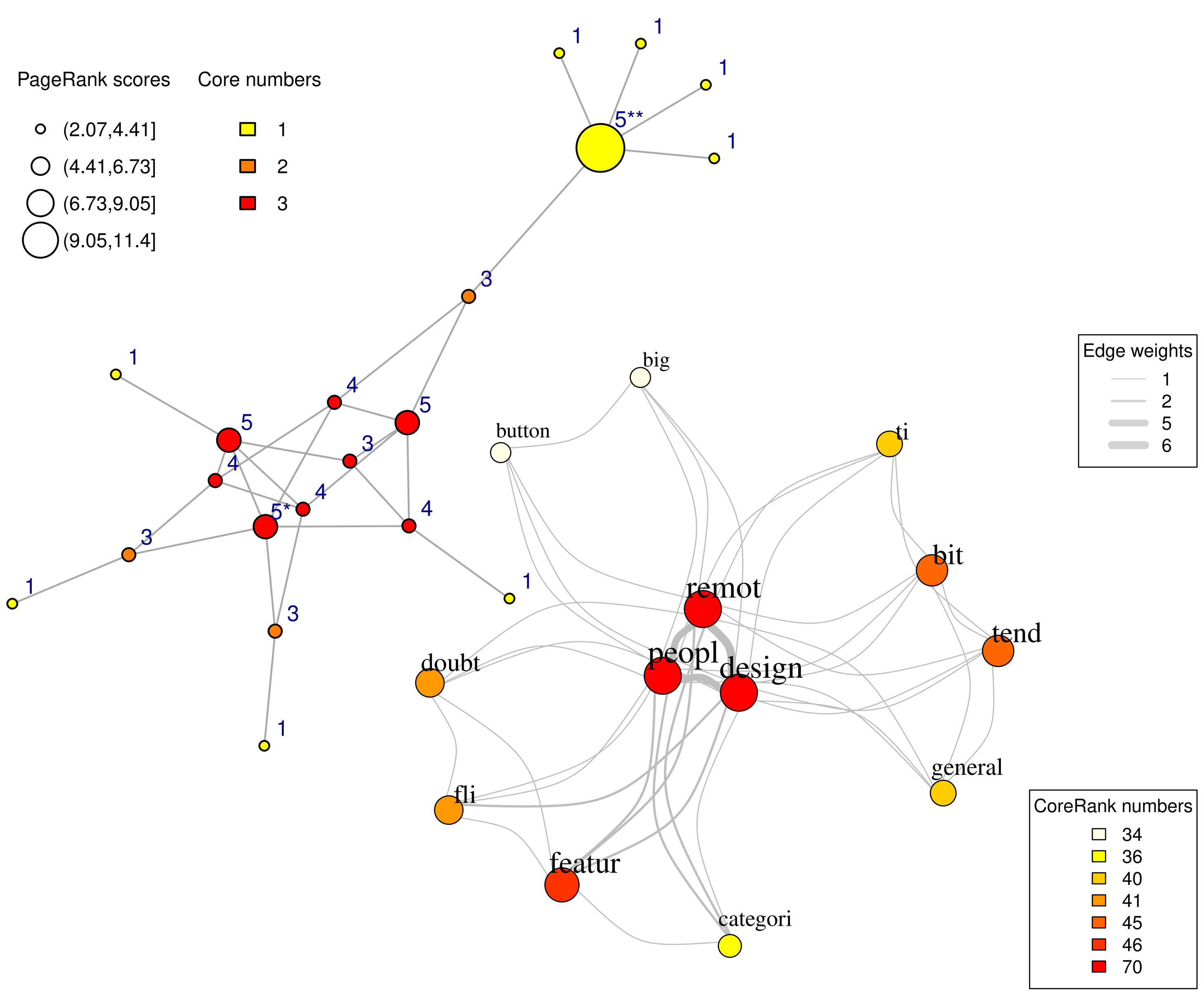


Figure 2: *k*-core decomposition and word co-occurrence graph

4. Submodular Maximization

Goal: generate the final summary by selecting an optimal subset S from the set of abstractive sentences $\mathcal{S}$ under a budget constraint.

$$\boxed{\underset{S \subseteq \mathcal{S}}{\operatorname{argmax}}\, f(S) | \sum_{s \in S} cost_s \leq budget}$$

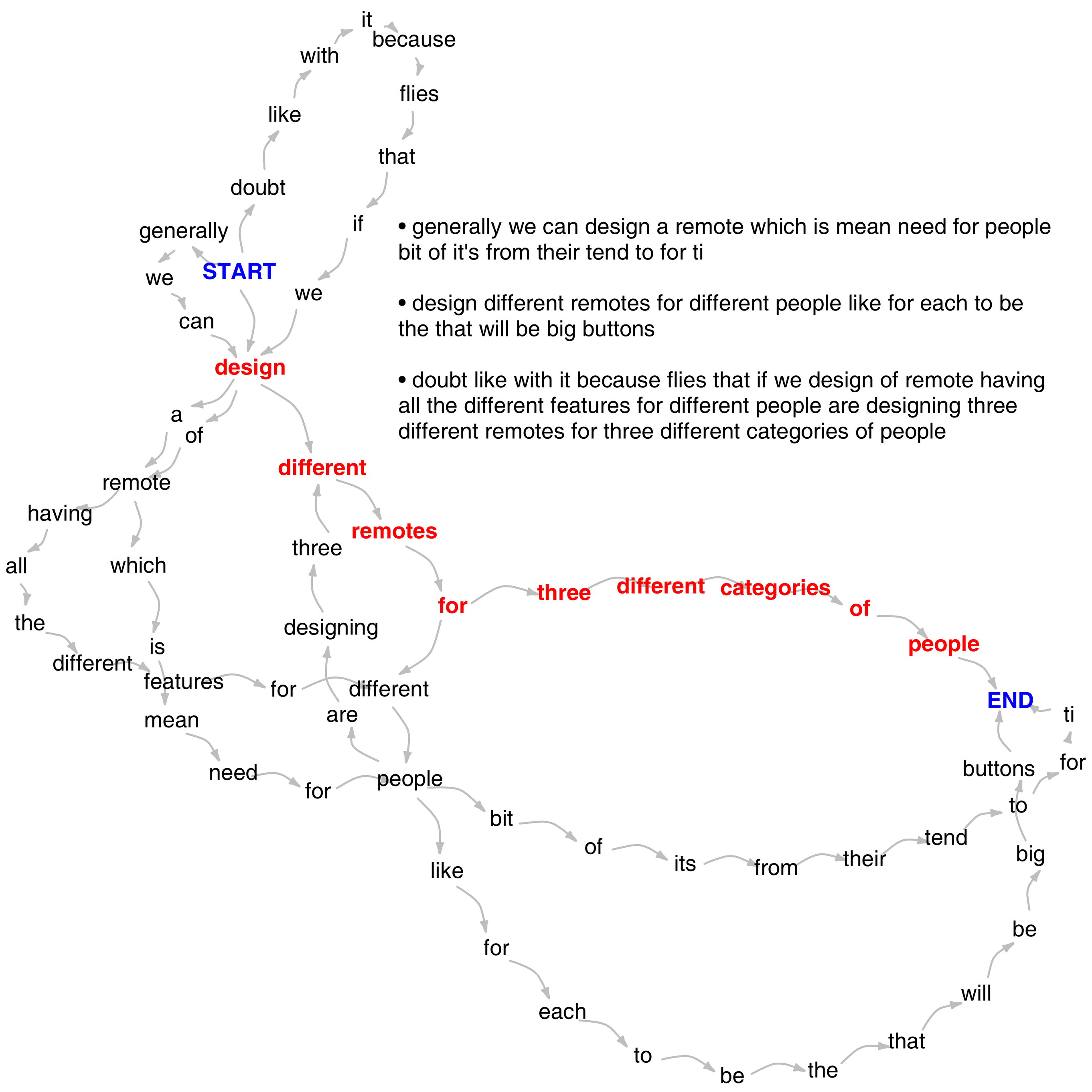
NP-hard, but near-optimal performance can be guaranteed with a modified greedy algorithm [Lin and Bilmes 2010] that iteratively selects the sentence s that maximizes the ratio of summary quality function gain to scaled cost $^{f(G \cup s) - f(G)}/_{cost_s^r}$ (where G is the current subset and $r \geq 0$ is a scaling factor).

Submodular and monotone non-decreasing quality function:

$$f(S) = \sum_{s_i \in S} n_{s_i} w_{s_i} + \lambda \sum_{j=1}^{k} 1_{\exists s_i \in S | s_i \in cluster_j}$$

$\lambda \geq 0$: trade-off parameter (coverage and diversity), n_{s_i}: number of occurrences of word s_i in S, w_{s_i}: CoreRank score of word s_i.

3. Multi-Sentence Compression (MSC)



Goal: generate an *abstractive sentence* for each *utterance community*, using an extension of the MSC graph [Filippova 2010, Boudin and Morin 2013, Mehdad et al. 2013].

Word Graph Building

⇒ Every input utterance is a loopless path ⇒ there are many other paths ⇒ goal: find the best compression path

Figure 3: MSC graph for a utterance community from AMI meeting IS1009b and a **possible compression path**.

Edge Weight Assignment

⇒ $\boxed{w'''(p_i, p_j) = {}^{w'(p_i,p_j)}/_{w''(p_i,p_j)}}$

▶ *Local co-occurrence statistics*:

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

Favors edges between words that frequently appear close to each other (word association).

$freq(p_i)$: number of words mapped to the node p_i.
$diff(P, p_i, p_j)^{-1}$: inverse of the distance between p_i and p_j in path P.

▶ *Global exterior knowledge* (Word Attraction Force [Wang et al. 2014]):

$$w''(p_i, p_j) = \frac{freq(p_i) \times freq(p_j)}{d^2_{p_i, p_j}}$$

Favor paths going through *salient* nodes that are *close* in the embedding space (*semantic relatedness*).

d_{p_i,p_j}: Euclidean distance of the word embedding vectors for p_i and p_j.

Path Reranking

⇒ $\boxed{W(P)/_{|P| \times F(P) \times C(P) \times D(P)}}$ ⇒ the lowest is the best compression path.

▶ The path with the lowest cumulative edge weight $W(P) = \sum_{i=1}^{|P|-1} w'''(p_i, p_{i+1})$ does not guarantee its readability nor informativeness ⇒ Reranking N best paths is necessary.
▶ Reranking strategy based on *Fluency*, *Coverage* and *Diversity*:

$$F(P) = \frac{\sum_{i=1}^{|P|} \log Pr(p_i | p_{i-n+1}^{i-1})}{\#n\text{-}gram} \quad C(P) = \frac{\sum_{p_i \in P} TW\text{-}IDF(p_i)}{\#p_i} \quad D(P) = \frac{\sum_{j=1}^{k} 1_{\exists p_i \in P | p_i \in cluster_j}}{|P|}$$

Results

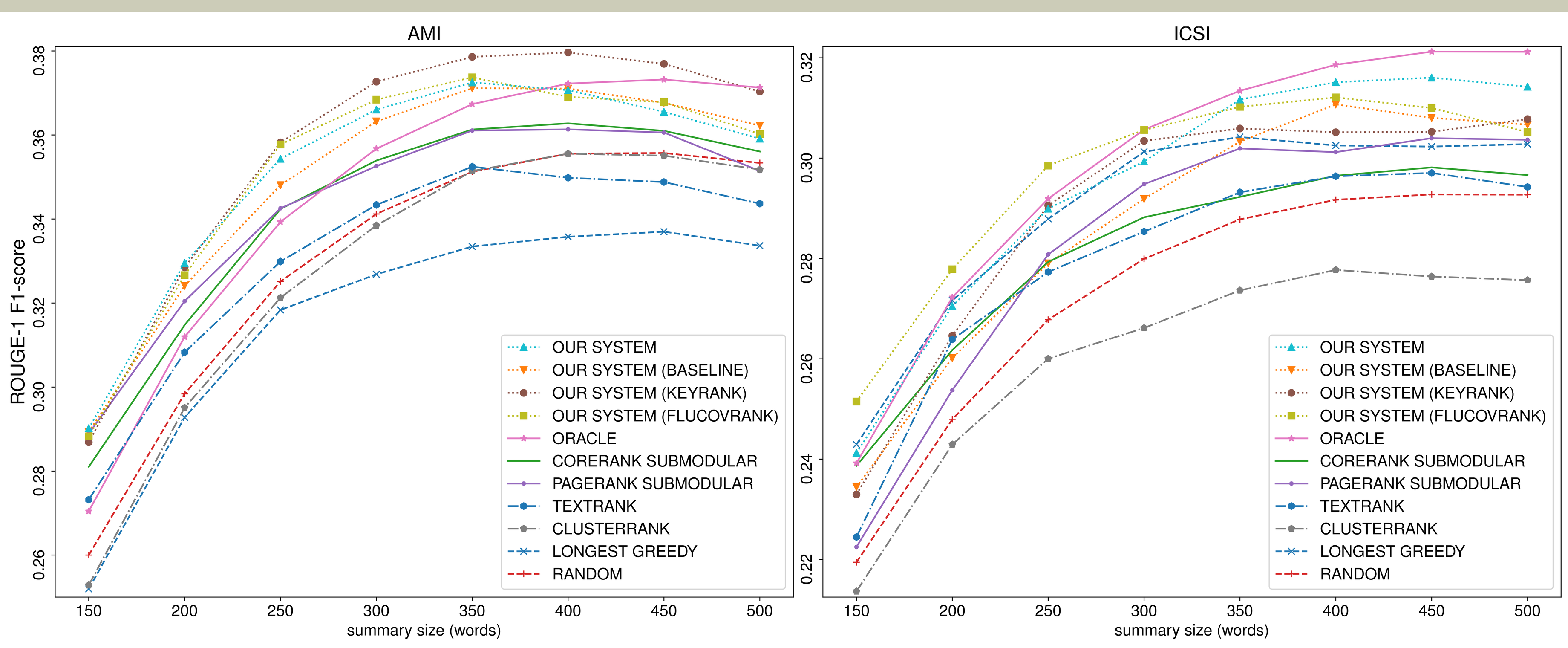


Figure 4: ROUGE-1 F-1 scores for various budgets (ASR transcriptions).

	AMI ROUGE-1			AMI ROUGE-2			AMI ROUGE-SU4			ICSI ROUGE-1			ICSI ROUGE-2			ICSI ROUGE-SU4		
	R	P	F-1	R	P	F-1	R	P	F-1	R	P	F-1	R	P	F-1	R	P	F-1
Our System	41.83	34.44	37.25	8.22	6.95	7.43	15.83	13.70	14.51	36.99	28.12	**31.60**	5.41	4.39	4.79	13.10	10.17	**11.35**
Our System (Baseline)	41.56	34.37	37.11	7.88	6.66	7.11	15.36	13.20	14.02	36.39	27.20	30.80	5.19	4.12	4.55	12.59	9.70	10.86
Our System (KeyRank)	42.43	35.01	**37.86**	8.72	7.29	**7.84**	16.19	13.76	**14.71**	35.95	27.00	30.52	4.64	3.64	4.04	12.43	9.23	10.50
Our System (FluCovRank)	41.84	34.61	37.37	8.29	6.92	7.45	16.28	13.48	14.58	36.27	27.56	31.00	5.56	4.35	**4.83**	13.47	9.85	11.29
Oracle	40.49	34.65	**36.73**	8.07	7.35	**7.55**	15.00	14.03	**14.26**	37.91	28.39	**32.12**	5.73	4.82	**5.18**	13.35	10.73	**11.80**
CoreRank Submodular	41.14	32.93	36.13	8.06	6.88	7.33	14.84	13.91	14.18	35.22	26.34	29.82	4.36	3.76	4.00	12.11	9.58	10.61
PageRank Submodular	40.84	33.08	36.10	8.27	6.88	7.42	15.37	13.71	14.32	36.05	26.69	30.40	4.82	4.16	4.42	12.19	10.39	11.14
TextRank	39.55	32.60	35.25	7.67	6.43	6.90	14.87	12.87	13.62	34.89	26.33	29.70	4.60	3.74	4.09	12.42	9.43	10.64
ClusterRank	39.36	32.53	35.14	7.14	6.05	6.46	14.34	12.80	13.35	32.63	24.44	27.64	4.03	3.44	3.68	11.04	8.88	9.77
Longest Greedy	37.31	30.93	33.35	5.77	4.71	5.11	13.79	11.11	12.15	35.57	26.74	30.23	4.84	3.88	4.27	13.09	9.46	10.90
Random	39.42	32.48	35.13	6.88	5.89	6.26	14.07	12.70	13.17	34.78	25.75	29.28	4.19	3.51	3.78	11.61	9.37	10.29

Table 1: Macro-averaged results for 350 and 450 word summaries (ASR transcriptions).

Supported in part by the project OpenPaaS::NG

ACL 2018

guokan.shang@polytechnique.edu