Exploiting local and global performance of candidate systems for aggregation of summarization techniques

Parth Mehta
IR&LP Lab
DA-IICT, India
`parth_me@daiict.ac.in`

Prasenjit Majumder
IR&LP Lab
DA-IICT, India
`p_majumder@daiict.ac.in`

Abstract

With an ever growing number of extractive summarization techniques being proposed, there is less clarity then ever about how good each system is compared to the rest. Several studies highlight the variance in performance of these systems with change in datasets or even across documents within the same corpus. An effective way to counter this variance and to make the systems more robust could be to use inputs from multiple systems when generating a summary. In the present work, we define a novel way of creating such ensemble by exploiting similarity between the content of candidate summaries to estimate their reliability. We define GlobalRank which captures the performance of a candidate system on an overall corpus and LocalRank which estimates its performance on a given document cluster. We then use these two scores to assign a weight to each individual systems, which is then used to generate the new aggregate ranking. Experiments on DUC2003 and DUC 2004 datasets show a significant improvement in terms of ROUGE score, over existing sate-of-art techniques.

1 Introduction

An ever increasing interest in the field of automatic summarization has led to a plethora of extractive techniques, with a lack of clarity about which is the best technique. Unclear implementation details and variation in evaluation setups only make the problem worse. However, there is no doubt that none of these techniques would always work. Not only will there be a difference in performance across different datasets, there is also a good amount of variation across documents in the same dataset. In several cases this difference is also attributed to the fact that different ROUGE setups are used for evaluation, which can result substantial variation in the scores. (Hong et al., 2014) propose using a fixed set of parameters for ROUGE and report comparable results for several summarization algorithms. Even with this normalization, the system performance still varies a lot, and there is a possibility of exploiting this variation to generate better performing systems. To give an example, we show a simple comparison of two extractive summarization systems from those used by (Hong et al., 2014) in their experiments. We pick two extreme systems, in terms of performance, from the those reported in the work. The FreqSum system(Nenkova et al., 2006), which has the weakest performance, was compared to the DPP system(Kulesza et al., 2012) which was the best performing system amongst those compared. On DUC 2004 dataset, FreqSum performed better on more than 10% of the document clusters. There are documents for which a system which is overall very weak, outperforms the system that has a very good performance on an average. Going a step further, an oracle of just the five baseline systems, outperforms DPP in a little over fifty percent of the document clusters. The argument is clear: ensemble of several systems can definitely improve the performance as compared to individual systems. The ideal way of forming an ensemble summary would be to select the relevant information from each candidate while discarding the rest.

In this work we propose a new method for estimating the authority of a particular system for a given document and at the same time also estimating the importance of each sentence within the summary generated by that system. The ensemble summary is then a function of the authority of each candidate

system as well as the relative importance of each sentence in the candidate summaries. The fact, that informative or *summary worthy* sentences in a document cluster are much less in number compared to the non-informative ones, forms the basis of our hypothesis. We argue that since this content is much less, any substantial overlap between two summaries will likely be due to the *important* content rather than the redundant one. Simply because there is less content to choose from when it comes to the important sentences two good summaries will have a good amount of overlap in content. At the same time it is highly unlikely that two summaries will also chose the same not-important content. Keeping this argument in mind we associate higher similarity in content between two summaries with the summaries having more informative content.

We use graph based ranking that takes into account the similarity of a candidate summary with other candidates to generate its local (or document specific) ranking. We also determine the overall global ranking of a system from its ROUGE score on a development dataset. In the same way *informativeness* of a sentence is linked to its overlap with sentences of other summaries. The HybridRank model proposed here, combines these three factors to generate a new aggregate ranking of sentences.

2 Related Work

In contrast to the amount of attention automatic summarization, and especially the extractive techniques, has achieved from researchers, aggregation techniques have been explored little. This is counter intuitive given several studies which show that even in cases where two systems achieve a comparable ROUGE score, the actual content can be quite different (Hong et al., 2014). Existing aggregation techniques can be broadly classified into two categories: *rank aggregation* and *summary aggregation*. These techniques are used post-summarization, i.e. each candidate system is first used without any modification. The output, whether ranked lists or summaries, are then used for aggregation. The former method solely relies on candidate summaries, without any information or assumption about the original sentence rankings. In contrast the latter combines existing ranked lists to generate a new aggregate ranking. Apart from these two, there is another type of aggregation which combines various aspects of candidates and incorporates them into the algorithm itself.

(Pei et al., 2012) and (Hong et al., 2015) are two instances of the summary aggregation techniques. (Pei et al., 2012) use SVM-Rank to learn the optimum ranking of sentences from candidate systems. Each sentence is labeled as $-1, 0, 1$ depending on its *summary worthiness* and then it is used to learn pairwise ranking for each sentence. As opposed to this, (Hong et al., 2015) attempt to generate rankings of entire summaries to find out the combination of sentences that maximizes ROUGE-1 or ROUGE-2. They use summaries from four different systems to begin with. Next they combine these summaries and list out all possible candidate summaries by selecting a fixed number of sentences from the combined set. Several word and summary level features are then used to train a SVM-Rank algorithm that can learn to rank candidate summaries to maximize ROUGE scores.

Excluding the common techniques like *Round Robin*, *Borda Count* or *Reciprocal Rank* only notable attempt at using rank aggregation was made by (Wang and Li, 2012). The *weighted consensus summarization* system proposed by them treats rank aggregation as a optimization problem. This approach also introduced the concept of consensus between candidate summaries. They create a weighted combination of ranked lists, under the constraint that the aggregate ranking be as close to original rankings as possible. Unlike the approach proposed in this paper, (Wang and Li, 2012) do not differentiate candidates based on their *trustworthiness*, and instead try to maintain as much information from each candidate as possible. Apart from not taking into account the content of candidate summaries, there are two major limitations with this approach. One, it uses L_1 norm for computing similarity (or distance) between candidate rankings, which can be a sub-optimal choice when compared to traditional metrics like *Kendall's Tau*. The other major problem is, this technique tries to optimize rankings over all sentences in the document cluster. So even if the systems agree on ranking of top-k sentence, which

actually go into the summary, the aggregation will try to optimize over the entire rankings. This is not only unnecessary, but can also affect the performance adversely.

Neither *summary aggregation* nor *rank aggregation* take into consideration the original summarization algorithms or in any way modify them. There have been a few attempts at modifying summarization algorithms to incorporate features from several candidates into a single algorithm. But such attempts are limited by the non-triviality of being able to meaningfully combine unique aspects of the candidates. One popular approach of this kind is combining several sentence similarity scores to generate an aggregate similarity which can then be used by any of the existing algorithms. The *MultSum* technique proposed by (Mogren et al., 2015) builds upon the submodular optimization technique(Lin and Bilmes, 2012). They replace the cosine similarity used in the original approach with multiplicative combination of several sentence similarity measures. This can be extended to several techniques which rely on a sentence similarity metric. But in general it is difficult to define an *aggregate* for other components of a summarization algortihm. As an alternate, (Mehta and Majumder, 2018) propose combining several aspects of candidate systems like *sentence similarity*, *ranking algorithm* and *text representation scheme* in a post-summarization setup. This approach falls within the rank aggregation technique, except that the candidate systems are created by varying one of these three components of the original systems. The *LexRank* algorithm can be used with Kullback Leibler Divergence or word overlap, instead of cosine similarity. Such variations are then combined using existing rank aggregation techniques.

3   Proposed Approach

In this paper we propose a new summary aggregation technique which takes into account the content of each candidate systems, rather than only ranked lists as done in (Wang and Li, 2012). In a multi document summarization setup, especially in case of newswire, it is quite common to have duplicate or near duplicate content getting repeated across multiple documents. To put things into perspective, DUC 2003 dataset has, on an average, 34 sentences per cluster which have an exact match within the document cluster. More than 50 sentences have a 80% match with another sentence. Most existing summarization techniques do not handle this redundancy explicitly, neither do most existing aggregation techniques. This has a huge impact on a class of aggregation techniques which rely only on ranked lists aggregation without taking into account the actual content of individual summaries. For instance, consider the example shown below where S_1 and S_2 are individual sentence rankings and S_A is the aggregate ranking. This would be fine if each sentence is different and equally important. But consider a case where $sim(s_1, s_4) = 1$. The fact that s_1 and s_5 are repeated across documents makes them more important. But the rank aggregation techniques fail to take into account their actual content and treat these separately, which may result in lowering of their aggregate scores. S_A^* indicates the ideal rank aggregate.
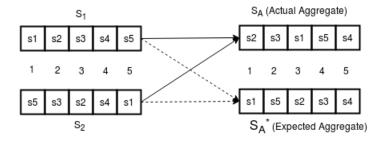

Figure 1: Issue with rank aggregation

Instead, in the proposed approach we take into account content of the summaries being aggregated to assign weights to candidate systems. The proposed system takes into account the overlap in content between summaries of the candidate systems to compute their reliability, and in turn use it to assign relative importance to individual sentences in those summaries.

The proposed method estimates importance of a given sentence in a particular candidate summary, and then uses this information to generate a meta-summary. In contrast to the solution proposed by (Hong et al., 2015), where they use word and summary level features to estimate the relative importance of each candidate, we propose using similarity between candidates as a measure for their importance. We argue that given two good summaries there is bound to be a good amount of overlap in their content, simply because the relevant content in a document cluster is much less compared to non-relevant content. Arguing the other way round, if two summaries have a good amount of overlap, it is likely because both have more *informative* content. Due to a large number of *non-informative* sentences, selecting the same set of *non-informative* sentences in two distinct summaries is much less likely. The only possible exception where two bad summaries might have higher overlap, will be when both have very similar sentence rankings. As the number of candidate systems increase, this will automatically be taken care of. We start with a simple method to estimate informativeness of sentences in candidate summaries.

3.1 SentRank

This system takes into account similarity of each sentence in a candidate summary, with that of other candidate summaries, and uses it to assign a relative importance to the sentence. The argument presented above, in favor of using similarity as a measure for reliability of a summary can easily be extended at sentence level. A sentence that is *summary worthy* will share more information with another *summary worthy sentences*.

For i^{th} sentence in the jth candidate summary(s_{ij}), we find the best matching sentence in the remaining candidate summaries. The score of that sentence can then be computed as shown in equation 1 below. Score of the sentence is the sum of its similarity with the best matching sentences from remaining candidates. Here j,k are the candidate systems. i and l are the sentences in candidate J and K respectively.

$$\boldsymbol{R}(s_{ij}) = \sum_{k,k \neq j} \max_l (Sim(s_{ij}, s_{lk})) \tag{1}$$

Each sentence in the candidate summary is then ranked according to their score $\boldsymbol{R}$ and top k sentences are selected in the summary. We experimented with n-gram overlap, cosine and KL Divergence for computing the similarity between sentences and empirically select cosine similarity, which is also used in the subsequent systems.

3.2 GlobalRank

One limitation of the SentRank approach proposed above is that does not take into account the reliability of candidate systems into account and treats each candidate equally. A sentence that comes from a well performing candidate system is more likely to be *informative* compared to a sentence from a poor summary. The proposed *GlobalRank* system does exactly that. It builds over the SentRank system by incorporating a candidates *global reputation* score into the sentence ranking scheme. The new scoring mechanism is shown in equation 2 below. $G(k)$ refers to the global reputation of candidate system k.

$$\boldsymbol{R}(s_{ij}) = \sum_{k,k \neq j} G(k) * \max_l (Sim(s_{ij}, s_{lk})) \tag{2}$$

$G(k)$ is estimated using the average ROUGE-1 recall of each candidate systems as shown in equation 4 and 5 below. $R1_k$ is the rouge-1 recall of the kth candidate. $R1'$ is the normalized version of $R1$. Here we do not subtract mean, to avoid negative values in scoring, and instead subtract the minimum of $R1'$. Additionally we scale it using a scaling factor a, which is dependent on the total number of candidate systems. We empirically set a to 0.1. We used the results on DUC2002 dataset for estimating the ROUGE-1 recall, and in turn the GlobalRank of a candidate.

$$G(k) = aR1'(k) \tag{3}$$

$$R1'_k = \frac{R1_k}{\sigma(R1_k)} - \min_k \left[\frac{R1_k}{\sigma(R1_k)} \right] \tag{4}$$

As compared to SentRank, which can be overwhelmed by too many poor performing systems, Global-Rank provides a smoothing effect, by giving more importance to the systems that are known to perform well generally.

3.3 LocalRank

One major limitation of the existing aggregation systems, which we highlighted in section 2, is their inability to predict which candidate system will perform better for a given document cluster. Neither of the systems suggested above, *SentRank* and *GlobalRank*, address this problem. The next system, *LocalRank* tries to mitigate this problem. We do not rely on any lexical or corpus specific features, simply because the training data is not sufficient to estimate these features reliably. Instead we continue on our line of argument, using the similarity between summaries as a measure of reliability. For a give document cluster, we estimate reliability of a candidate k from the content it shares with other candidates, and also the reliability of those candidates. We first create a graph with the nodes as the candidate summaries and edge as the similarity between nodes. Each candidate starts with the same reputation score or LocalRank (L). The local rank is then updated iteratively using the pagerank algorithm (Page et al., 1999). The Local rank for a given node is estimated as shown in equation 5 below:

$$L(k) = \sum_j L(j) * Sim(S_j, S_k) \tag{5}$$

$L(k)$ indicates local rank of kth candidate, S_k indicate summary generated by the kth candidate. We use cosine similarity as the similarity score. The overall sentence scores are then computed just like in the GlobalRank algorithm.

$$\boldsymbol{R}(s_{ij}) = \sum_{k,k \neq j} L(k) * \max_l (Sim(s_{ij}, s_{lk})) \tag{6}$$

3.4 HybridRank

While LocalRank is useful for estimating how well a given candidate might perform for a given document cluster, it does not make use of the actual system performance. HybridRank overcomes that limitation. As the name suggests, HybridRank combines strengths of both GlobalRank as well as LocalRank by taking a weighted combination of both. The HybridRank is defined in the equation 7 below

$$H(k) = \alpha L(k) + (1 - \alpha)G(k) \tag{7}$$

Here the value of α determines the balance between Local and GlobalRank. Higher value of Alpha gives more importance to the estimate of how good a system will perform on a particular cluster, while ignoring the overall aggregate performance of candidate. $\alpha = 0$ leads to the original GlobalRank, without any local information. We empirically set the value of α to 0.3. Once the systems are ranked, the sentence rankings are computed in the same manner as LocalRank or GlobalRank (equation 2 and 6).

4 Experimental Setup and Results

We report the experimental results on the DUC 2003 and DUC 2004 datasets. We report standard ROUGE scores(Lin, 2004) that are well accepted across the community, ROUGE-1, ROUGE-2 and ROUGE-4 recall. To be consistent and in order to make our work reproducible, we use the same set of ROUGE parameters as that used by (Hong et al., 2014)[1]. We use eleven candidate systems which are a

---

[1]ROUGE-1.5.5.pl -n 4 -m -a -x -l 100 -c 95 -r 1000 -f A -p 0.5 -t 0

mix of several state of art extractive techniques and other well known baseline systems. The complete list is shown below, with a brief description of each system. For the DUC 2004 dataset, (Hong et al., 2014) provide summaries for all these systems. We directly use these pre-generated summaries provided for that particular year, to provide a fair comparison. We generated results for the DUC 2003 dataset ourselves. We do post-processing on the generated summaries, by dropping sentences that have a cosine similarity of more than 0.5 with the already selected sentences. Apart from that no other pre/post-processing was done.

We use the following systems as candidate systems for our experiments:

LexRank This method proposed by (Erkan and Radev, 2004) treats each document as a undirected graph and each sentence in the document constitutes a node. The edges represent cosine similarity between nodes. The importance of each node is then iteratively determined by the number of other nodes to which it is connected and the importance of those nodes.

FreqSum A simple approach(Nenkova et al., 2006) that ranks each sentence based on the frequency of its constituent words. Higher the frequency more the informativeness of the word.

TsSum This method defines *topic signatures* as the words which are more frequent in a given document compared to a background corpus(Lin and Hovy, 2000). Sentenced with more *topic signatures* are considered more informative

Greedy-KL This method follows a greedy algorithm(Haghighi and Vanderwende, 2009) to minimize the KL Divergence between the original document and resultant summary. Sentences are sequentially added to the summary so as to minimize the KL-Divergence between word distributions of the set of sentences selected so far and the overall document

CLASSY04 Judged best among the submissions at DUC 2004(Conroy et al.,), uses a hidden markov model with topic signatures as the features. It links the usefulness of a sentence to that of its neighboring sentences.

CLASSY11 This method builds over the CLASSY04 technique, and uses topic signatures as features while estimating the probability that a bigram will occur in human generated summary. It employs non negative matrix factorization to select a subset of non-redundant sentences with highest scores.

Submodular (Lin and Bilmes, 2012) treat summarization as a submodular maximization problem. It incrementally computes the *informativeness* of a summary and also provides a confidence score as to how close the approximation is to globally optimum summary.

DPP Detrimental point processing(Kulesza et al., 2012) is the best performing state-of-art system amongst all the candidate systems. DPP scores each sentence individually, while at the same time trying to maintain a global diversity to reduce redundancy in the content selected.

RegSum uses diverse features like parts of speech tags, name entity tags, locations and categories for supervised prediction of word importance(Hong and Nenkova, 2014). The sentence with most number of *important* words are then included in the summary.

OCCAMS_V The system by (Davis et al., 2012), employs LSA to estimate word importance and then use the budgeted maximal coverageand the knapsack problem to generate sentence rankings.

ICSISumm treats summarization as a global linear optimization problem(Gillick et al.,), to find globally best summary instead of selecting sentences greedily. The final summary includes most important concepts in the documents.

	DUC2003			DUC2004		
System	R-1	R-2	R-4	R-1	R-2	R-3
LexRank	0.3572	0.0742	0.0079	0.3595	0.0747	0.0082
FreqSum	0.3542	0.0815	0.0101	0.3530	0.0811	0.0099
TsSum	0.3589	0.0863	0.0103	0.3588	0.0815	0.0103
Greedy-KL	0.3692	0.0880	0.0129	0.3780	0.0853	0.0126
CLASSY04	0.3744	0.0902	0.0148	0.3762	0.0895	0.0150
CLASSY11	0.3730	0.0925	0.0142	0.3722	0.0920	0.0148
Submodular	0.3888	0.0930	0.0141	0.3918	0.0935	0.0139
DPP	0.3992	0.0958	0.0159	0.3979	0.0962	0.0157
RegSum	0.3840	0.0980	0.0165	0.3857	0.0975	0.0160
OCCAMS_V	0.3852	0.0976	0.0142	0.3850	0.0976	0.0133
ICSISumm	0.3855	0.0977	0.0185	0.3840	0.0978	0.0173
Borda Count	0.3700	0.0738	0.0115	0.3772	0.0734	0.0110
WCS	0.3815	0.0907	0.0120	0.3800	0.0923	0.0125
SentRank	0.3880	0.1010	0.0163	0.3870	0.1008	0.0159
GlobalRank	0.3562	0.1045	0.0185	0.3955	0.1039	$\mathbf{0.0191}^{\dagger}$
LocalRank	0.3992	0.1058	0.0192	0.3998	0.1050	0.0187
HybridRank	$\mathbf{0.4082}^{\dagger}$	$\mathbf{0.1102}^{\dagger}$	$\mathbf{0.0195}^{\dagger}$	$\mathbf{0.4127}^{\dagger}$	$\mathbf{0.1098}^{\dagger}$	0.0180

Table 1: Results on DUC 2003 dataset

As benchmark ensemble techniques, we use two existing techniques: *Borda count* and *Weighted consensus summarization*(Wang and Li, 2012) described in section 2. For the GlobalRank system, we used DUC 2002 dataset as a development dataset to estimate the overall performance of candidate systems. We ranked the systems based on ROUGE-1 recall scores for this purpose. The results are shown in table 1 below:

As shown in the table, the proposed systems outperform most existing systems on all three ROUGE scores. Both Borda and WCS failed to outperform the best state of art results. Even the simplistic SentRank algorithm outperforming most candidate systems and achieves a performance at par with the state of art systems in terms of ROUGE-2. While HybridRank achieves the best performance for ROUGE-1 and ROUGE-2 on both DUC 2003 and DUC 2004 datasets, GlobalRank performs best in terms of ROUGE-4 on DUC 2004. We performed a two sided sign test for determining whether the results were significantly different. The results clearly show that a rank aggregation technique that takes into account content of the summaries achieve a much higher ROUGE score vis-a-vis the systems that use only the ranked lists of sentences.

5 Conclusion

In this work we propose a new technique to estimate *reliability* of a summarization system for a given document cluster. We define three systems, *SentRank*, *LocalRank* and *GlobalRank* which take into account *informativeness* of individual sentences, performance of candidates on a given document cluster, and overall performance of candidates on a held out development set, respectively. We use content overlap between summaries generated from several systems to estimate the relative importance of each system in case of LocalRank and SentRank. We combine the information from all these three systems to generate the final hybrid ranking (HybridRank) system. Summaries generated from such an aggregate system outperforms all the baseline and state of art systems as well as the baseline aggregation techniques by a good margin.

References

[Conroy et al.] John M Conroy, Judith D Schlesinger, Jade Goldstein, and Dianne P Oleary. Left-brain/right-brain multi-document summarization. In *n Proceedings of the Document Understanding Conference (DUC) 2004.*

[Davis et al.2012] Sashka T Davis, John M Conroy, and Judith D Schlesinger. 2012. Occams–an optimal combinatorial covering algorithm for multi-document summarization. In *Data Mining Workshops (ICDMW), 2012 IEEE 12th International Conference on*, pages 454–463. IEEE.

[Erkan and Radev2004] Günes Erkan and Dragomir R Radev. 2004. Lexrank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479.

[Gillick et al.] Daniel Gillick, Benoit Favre, Dilek Hakkani-Tür, Bernd Bohnet, Yang Liu, and Shasha Xie. The icsi/utd summarization system at tac 2009.

[Haghighi and Vanderwende2009] Aria Haghighi and Lucy Vanderwende. 2009. Exploring content models for multi-document summarization. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 362–370. Association for Computational Linguistics.

[Hong and Nenkova2014] Kai Hong and Ani Nenkova. 2014. Improving the estimation of word importance for news multi-document summarization. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 712–721.

[Hong et al.2014] Kai Hong, John M Conroy, Benoit Favre, Alex Kulesza, Hui Lin, and Ani Nenkova. 2014. A repository of state of the art and competitive baseline summaries for generic news summarization. In *Proceedings of Language Resources and Evaluation Conference*, pages 1608–1616.

[Hong et al.2015] Kai Hong, Mitchell Marcus, and Ani Nenkova. 2015. System combination for multi-document summarization. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 107–117, Lisbon, Portugal, September. Association for Computational Linguistics.

[Kulesza et al.2012] Alex Kulesza, Ben Taskar, et al. 2012. Determinantal point processes for machine learning. *Foundations and Trends in Machine Learning*, 5(2–3):123–286.

[Lin and Bilmes2012] Hui Lin and Jeff Bilmes. 2012. Learning mixtures of submodular shells with application to document summarization. In *Proceedings of the Twenty-Eighth Conference on Uncertainty in Artificial Intelligence*, pages 479–490. AUAI Press.

[Lin and Hovy2000] Chin-Yew Lin and Eduard Hovy. 2000. The automated acquisition of topic signatures for text summarization. In *Proceedings of the 18th conference on Computational linguistics-Volume 1*, pages 495–501. Association for Computational Linguistics.

[Lin2004] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: Proceedings of the ACL-04 workshop*, volume 8. Association for Computational Linguistics.

[Mehta and Majumder2018] Parth Mehta and Prasenjit Majumder. 2018. Effective aggregation of various summarization techniques. *Information Processing & Management*, 54(2):145–158.

[Mogren et al.2015] Olof Mogren, Mikael Kågebäck, and Devdatt Dubhashi. 2015. Extractive summarization by aggregating multiple similarities. In *Proceedings of Recent Advances In Natural Language Processing*, pages 451–457.

[Nenkova et al.2006] Ani Nenkova, Lucy Vanderwende, and Kathleen McKeown. 2006. A compositional context sensitive multi-document summarizer: exploring the factors that influence summarization. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 573–580. ACM.

[Page et al.1999] Lawrence Page, Sergey Brin, Rajeev Motwani, and Terry Winograd. 1999. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford InfoLab.

[Pei et al.2012] Yulong Pei, Wenpeng Yin, Qifeng Fan, and Lian'en Huang. 2012. A supervised aggregation framework for multi-document summarization. In *Proceedings of 24th International Conference on Computational Linguistics: Technical Papers*, pages 2225–2242.

[Wang and Li2012] Dingding Wang and Tao Li. 2012. Weighted consensus multi-document summarization. *Information Processing & Management*, 48(3):513–523.