Unsupervised Query-Focused Multi-Document Summarization using the Cross Entropy Method

Guy Feigenblat, Haggai Roitman, Odellia Boni, David Konopnicki
IBM Research - Haifa, Haifa, Israel,31905
guyf,haggai,odelliab,davidko@il.ibm.com

ABSTRACT

We present a novel unsupervised query-focused multi-document summarization approach. To this end, we generate a summary by extracting a subset of sentences using the Cross-Entropy (CE) Method. The proposed approach is generic and requires no domain knowledge. Using an evaluation over DUC 2005-2007 datasets with several other state-of-the-art baseline methods, we demonstrate that, our approach is both effective and efficient.

1 INTRODUCTION

The abundance of unstructured information raises the need for automatic systems that can "condense" information from various documents into a shorter length, readable summary. Such summaries may further be required to cover a specific information need (e.g., summarizing web search results).

Various methods have been proposed for the query-focused summarization task. These methods can be categorized based on two main dimensions [7], namely: *extractive* vs. *abstractive* and *supervised* vs. *unsupervised*. Extractive methods generate a summary using only text fragments extracted from the document(s). Compared to that, abstractive methods may also synthesize new text.

Supervised methods try to fit a model that learns to select or generate "relevant" text fragments for a summary based on training data. While supervised methods may provide better quality, they require more domain-knowledge compared to their unsupervised counterparts. Therefore, generalizing supervised methods to new datasets, domains, languages, etc, still remains a great challenge [7].

Some summarization methods further employ sentence compression steps so as to improve the summarization quality. Yet, such compression is commonly done using manually crafted rules or additional learning which requires ever deeper domain and linguistic knowledge [7].

We present a novel **extractive** and **unsupervised** query-focused multi-document summarization approach. Our approach requires **no domain knowledge**. Within our approach, summaries are solely generated by extracting a subset of sentences from the documents' text. We first present a generic solution to the sentence subset selection problem based on the *Cross-Entropy* (CE) *Method* [8]. We then suggest a specific instantiation based on a simple combination of several query-dependent and query-independent features. Using an evaluation over DUC 2005-2007 datasets with several other state-of-the-art baseline methods, we demonstrate that, our approach can be both effective and efficient.

2 APPROACH

Let q denote a given query, $\mathcal{D}$ denote a set of one or more matching documents to summarize and L be the constraint on the summary length. Our goal is to find a query-relevant and length-compliant summary S extracted from $\mathcal{D}$.

2.1 Sentence subset selection

We focus on sentences as the smallest possible extractable text units. Therefore, we assume that each document $D \in \mathcal{D}$ is represented by the sequence of sentences it contains. Our extractive solution is based on a *sentence subset selection* approach.

Let $\mathcal{S}_{\mathcal{D}}$ denote the set of all sentences in $\mathcal{D}$'s documents. Let $len(x)$ denote the length (in words count) of text x (e.g., $len(s)$ denotes the length of a given sentence $s \in \mathcal{S}_{\mathcal{D}}$). Furthermore, for a given subset $S \subseteq \mathcal{S}_{\mathcal{D}}$, representing a potential summary, let $len(S) = \sum_{s \in S} len(s)$ and $|S|$ denote its (summary) length and the number of sentences it contains, respectively.

We now cast the selection problem as a constrained global optimization problem. Our goal is, therefore, to find a subset of sentences $S \subseteq \mathcal{S}_{\mathcal{D}}$ that maximizes a given quality target function $Q(S|q, \mathcal{D})$ and satisfies $len(S) \leq L$.

2.2 Cross Entropy Summarizer

We now describe our summarization approach, hereinafter named *Cross Entropy Summarizer* (**CES** for short). **CES** utilizes an **unsupervised** approach to the sentence subset selection problem based on the *Cross Entropy Method* [8]. The CE-Method is a generic Monte-Carlo framework for solving hard combinatorial optimization problems based on rare event estimation [8]. Using this method allows to evaluate various instantiations of quality target functions $Q(S|q, \mathcal{D})$. On the concrete side, we shall next propose an actual instantiation of $Q(S|q, \mathcal{D})$, tailored to our summarization problem.

Using the CE-Method, we learn a (global) optimal selection policy as follows. For a given sentence $s \in \mathcal{S}_{\mathcal{D}}$, let $\varphi(s)$ denote the odds of it being selected for the summary S. In the beginning, each sentence has an equal chance of being selected (i.e.: $\varphi_0(s) = 0.5$). In the end of the CE-Method run we shall obtain the optimal selection policy $\varphi^*(\cdot)$. The optimal selection policy is iteratively learned as follows. At each iteration $t = 1, 2, \ldots$, we first sample N subsets $S_j \subseteq \mathcal{S}_{\mathcal{D}}$ according to the selection policy $\varphi_{t-1}(\cdot)$. Each subset S_j is sampled by independently choosing each sentence $s \in \mathcal{S}_{\mathcal{D}}$ to be

SIGIR'17, August 7–11, 2017, Shinjuku, Tokyo, Japan
© 2017 ACM. 978-1-4503-5022-8/17/08…$15.00
DOI: http://dx.doi.org/10.1145/3077136.3080690

included in S with probability $\varphi_{t-1}(s)$. Policy $\varphi_t(\cdot)$ is then derived by applying the following update rule:

$$\varphi_t(s) = \frac{\sum_{j=1}^{N} \delta_{[Q(S_j|q,\mathcal{D}) \geq \gamma_t]} \delta_{[s \in S_j]}}{\sum_{j=1}^{N} \delta_{[Q(S_j|q,\mathcal{D}) \geq \gamma_t]}}, \tag{1}$$

where $\delta_{[c]}$ denotes the *Kronecker-delta* function, having the value of 1 if condition c is satisfied (else 0). For a given $\rho \in (0,1]$, γ_t further denotes the $(1-\rho)$-quantile of the $Q(S_j|q,\mathcal{D})$ sample performances, obtained by sorting the sample according to $Q(S_j|q,\mathcal{D})$. Hence, according to Eq. 1, the selection likelihood of a given sentence $s \in \mathcal{S}_{\mathcal{D}}$ is proportional to its occurrence in the ρ-sample subsets S_j with the *highest performance* $Q(S_j|q,\mathcal{D})$ in iteration t; termed the *Elite sample* [8]. In order to handle the constraint on summary length L, whenever a sample S_j violates the constraint (i.e., $len(S_j) > L$), we simply assign $Q(S_j|q,\mathcal{D}) = -\infty$. To allow better tradeoff between *exploitation* (i.e., $\varphi_{t-1}(\cdot)$) and *exploration* (i.e., $\varphi_t(\cdot)$), following [8], the selection policy $\varphi_t(\cdot)$ is further smoothed as follows: $\varphi_t(\cdot)' = \alpha\varphi_{t-1}(\cdot) + (1-\alpha)\varphi_t(\cdot)$; with $\alpha \in [0,1]$. The CE Method runs until convergence (e.g., γ_t values do not change anymore for several consecutive iterations [8]). In the end of its run, we generate the summary by simply sampling a single subset $S^* \subseteq \mathcal{S}_{\mathcal{D}}$ according to $\varphi^*(\cdot)$ (the final policy learned at termination time). Upon termination, the CE Method guarantees convergence to the global optimal (or near-optimal) solution [8].

2.3 $Q(S|q,\mathcal{D})$ instantiation

We next propose a concrete instantiation of $Q(S|q,\mathcal{D})$. Since we assume an **unsupervised** learning setting, the actual instantiations of $Q(S|q,\mathcal{D})$ should correlate as much as possible with the actual (**unknown**) summary quality. Moreover, since we assume that every feasible sample is inherently a plausible candidate summary, we face a *one-class learning* problem. In this work, we propose to solve this problem using a feature engineering approach with m features. To this end, each feature $Q_i(S|q,\mathcal{D})$ we utilize ($1 \leq i \leq m$), is assumed to positively correlate with the actual quality; hence, we wish to maximize $Q_i(S|q,\mathcal{D})$.

Aiming at maximizing several such features together, we solve the following optimization problem:

$$\max_{S \subseteq \mathcal{S}_{\mathcal{D}} \wedge len(S) \leq L} Q_{CombMult}(S|q,\mathcal{D}), \tag{2}$$

where $Q_{CombMult}(S|q,\mathcal{D}) = \prod_{i=1}^{m} Q_i(S|q,\mathcal{D})$ is the **CombMult** fusion function [12]. The solution is obtained using the CE-Method as was described above.

2.4 Summarization features

We conclude this section with a description of six different features $Q_i(S|q,\mathcal{D})$ that we utilize in our approach.

The first two features estimate to what extent each candidate summary (subset) $S \subseteq \mathcal{S}_{\mathcal{D}}$ covers the information need expressed in query q. To this end, let $p_x^{[\mu]}(w) \overset{def}{=} \frac{tf(w,x)+\mu\frac{tf(w,C)}{len(C)}}{len(x)+\mu}$ denote the *Dirichlet* smoothed language model (LM) of text x with parameter μ and C further denote a given background corpus (e.g., Wikipedia). Our first feature, $Q_1(S|q,\mathcal{D}) = \sum_{w \in q} \sqrt{p_q^{[0]}(w) \cdot p_S^{[\mu]}(w)}$, measures the *Bhattacharyya* similarity (coefficient) between the unigram LM of query q and the unigram LM of summary S[1]. The second feature, $Q_2(S|q,\mathcal{D}) = \sum_{w \in q} p_S^{[0]}(w)$, simply measures the relative mass that summary S "devotes" to the query.

Our third feature (salience), $Q_3(S|q,\mathcal{D}) = \frac{\vec{S} \cdot \vec{\mathcal{D}}}{\|\vec{S}\|\|\vec{\mathcal{D}}\|}$ measures to what extent the summary S (generally) covers the document set $\mathcal{D}$. For that, we represent both S and $\mathcal{D}$ as TF-IDF term vectors. Here we use only bigrams, which commonly represent more important content units (e.g., named entities, concepts, etc).

Our fourth feature (diversity), $Q_4(S|q,\mathcal{D}) = -\sum_{w \in S} p_S^{[0]}(w) \log p_S^{[0]}(w)$, measures the summary's diversity by calculating the (bigram LM) entropy of summary S. The higher the entropy is, the more aspects are covered, and therefore, the higher the diversity of the summary is expected to be.

Our next feature (position) is $Q_5(S|q,\mathcal{D}) = \sqrt[|S|]{\prod_{s \in S}\left(1 + \frac{1}{\log(2+pos(s))}\right)}$, where $pos(s)$ is the relative start position of sentence s in its containing document D_s. This feature biases sentence selection towards sentences that appear earlier in their containing documents.

Our last feature (length), $Q_6(S|q,\mathcal{D}) = \frac{1}{|S|} len(S)$, further biases the selection towards longer summaries (closer to the length constraint) that contain few long sentences rather than summaries that contain many short ones. Therefore, the higher $Q_6(S|q,\mathcal{D})$, the more informative the summary is expected to be.

3 EVALUATION

3.1 Experimental Setup

3.1.1 Datasets. Our evaluation is based on the *Document Understanding Conferences* (DUC) 2005, 2006 and 2007 benchmarks[2]. Each benchmark contains a set of topic statements, each statement is associated with a set of English news articles (documents). Topic statements in the DUC benchmarks can be quite complex and include the main topic followed by one or more additional questions that elaborate what topic aspects should the summary cover; e.g.:

> *"Art and music in public schools. Describe the state of teaching art and music in public schools around the world. Indicate problems, progress and failures".*

The main task is, given a pair of (topic statement, documents), to generate a fluent, well-organized 250-word summary (i.e., $L = 250$) of the documents that answers the question(s) in the topic statement [1].

Overall, both DUC 2005 and 2006 benchmarks include 50 topic statements, while the DUC 2007 benchmark includes 45 topic statements. Furthermore, in DUC 2006 and 2007 each topic statement has 25 documents to be summarized and in DUC 2005 each topic statement has 32 documents. The DUC documents are pre-segmented into sentences (by NIST). Our approach, therefore, considers each sentence from the documents as a potential candidate for the summary. We further processed the text of topic statements and documents (i.e., tokenization, lowercasing, stopping, stemming, etc) using Lucene's English text analysis[3].

[1] Obtained by concatenating the text of the sentences in S.

[2] http://www-nlpir.nist.gov/projects/duc/data.html

[3] https://lucene.apache.org/core/641/analyzers-common/index.html

3.1.2 Query preprocessing. To handle DUC's complex querying scenario, we first generated for each topic statement $k \geq 1$ queries; each such query was generated by concatenating the main topic's text to the text of a given elaborating question. Trying to maximize the information match between queries and candidate sentences, we further utilized a query expansion approach. To this end, we represent each sub-query by the top-100 expansion (unigram) words obtained from a Wikipedia corpus [13]. The set of expanded queries was then used for calculating our query-depended objectives by summing over all queries, i.e.: $Q_i(S|q, \mathcal{D}) = \sum_{l=1}^{k} Q_i(S|q_l, \mathcal{D}); i \in \{1, 2\}$, with $\mu = 1000$.

3.1.3 Evaluation measures. For each topic statement, four human generated summaries (serving as the ground truth) are provided for quality evaluation [1]. To measure the summary quality, we adopt the ROUGE metric [5], which is the DUC main evaluation method [1]. We use the official ROUGE 1.5.5 toolkit with the standard parameters setting[4]. We report both Recall and F-measure of ROUGE-1, ROUGE-2 and ROUGE-SU4; where F-measure is the harmonic mean of ROUGE-Recall and ROUGE-Precision [5]. ROUGE-1 and ROUGE-2 measure the overlap in unigrams and bigrams between the candidate and the reference summaries, respectively [5]. ROUGE-SU4 further measures the overlap in skip-grams separated by up to four words [5].

3.1.4 CES implementation. We implemented **CES** in Java 1.8 on Windows 7, 8-core and 16GB memory. We note that, since sentences are independent of each other in our approach, all computation steps in **CES** (i.e., sampling, objective evaluation, sample sorting, and policy update) can be efficiently implemented using parallel computing [2]. Using such an implementation, the "heaviest" computation step is the sampling of a single sample $S \subseteq \mathcal{S}_{\mathcal{D}}$ and its objective $Q(S|q, \mathcal{D})$ calculation, an order of $O(|\mathcal{S}_{\mathcal{D}}|)$.

To speedup the computation, we further implemented an extended version of **CES**, which applies a preliminary step of sentence pruning. To this end, the pruned **CES** version only considers the top-l ($\in \{50, 100\}$) sentences in $\mathcal{S}_{\mathcal{D}}$ with the highest (unigram) *Bhattacharyya* similarity to the topic statement's expanded queries.

Finally, following previous recommendations [8], **CES** hyperparameters were fixed as follows: $N = 10,000$, $\rho = 0.01$ and $\alpha = 0.7$.

3.2 Baselines

We compare **CES** with several other state-of-the-art summarization methods which are also **unsupervised** (or at least require minimum learning), as follows:

- **BI-PLSA** [9] is a variant of the *probabilistic latent semantic analysis* (PLSA) method that simultaneously clusters and summarizes documents.
- **HierSum** [3] uses a hierarchical LDA-style model that represents content specificity as a hierarchy of topic vocabulary distributions.
- **MultiMR** [10] is a graph based multi-modality learning system that considers within and cross document sentence relationships. We compare only against the *linear fusion scheme* which achieved the best performance [10].
- **SpOpt-Δ** [14] selects as the summary the subset of sentences that minimizes the *documents reconstruction error*, having documents represented by sparse coding. It is built on top of the **DSDR** algorithm [4] by adding a sentence dissimilarity term to the objective function to encourage diversity. **SpOpt-Δ** also has a variant that utilizes sentence compression. Yet, since we do not apply a similar step, we only compare against the base **SpOpt-Δ** approach. Furthermore, we do not report the results of DSDR since it's performance is dominated by **SpOpt-Δ** [14].
- **DocRebuild** [6] also minimizes the documents reconstruction error using a *neural document model* on top of DSDR and **SpOpt-Δ** (with compression) methods. **DocRebuild** utilizes two document representations, namely: *Bag-of-Words* (BoW) model that represents each document term as a bag of words; and the *Paragraph Vector* (PV) model that considers words order. While the PV model achieves better performance compared to BoW, its main drawback is that, words representations are learned on DUC 2006-2007 datasets. Therefore, the quality depends on receiving all input documents for training beforehand. Here, we only report on the **DocRebuild** variants on top of DSDR, which do not utilize sentence compression.
- **QODE** [15] is a deep learning multi-document summarization approach that combines *Restricted Bolzmann Machines* (RBM) and *dynamic programming*.
- **CTSUM** [11] incorporates uncertainty in summarization by automatically predicting sentence uncertainty and incorporating it in a graph-based ranking scheme.

While some of the above baselines (**SpOpt-Δ**, **DocRebuild** and **CTSUM**) are presumed to be unsupervised, they still tuned few parameters using the DUC 2005 dataset. Therefore, for these baselines only results on the DUC 2006-2007 benchmarks were reported. Compared to that, **CES** is completely unsupervised, as we do not try to tune any parameter. Therefore, we report our evaluation results on all benchmarks. We note again that, for both the query expansion step and CE-Method run we basically reused recommended parameter settings [8, 13]. Moreover, some baselines only report on Recall-ROUGE while others only report on F-Measure ROUGE. To compare with all baselines, we report on both versions of ROUGE.

We further note that, since **CES** is a stochastic optimization approach, some deviation in summarization quality between runs is possible. Hence, on each benchmark, to measure the variability in quality, for each topic statement, we run **CES** (and its pruned version) 30 times and report on the average performance and its 95%-confidence interval. Furthermore, on each **CES** run, per topic statement, we recorded the number of iterations t and the absolute runtime until its convergence.

3.3 Results

3.3.1 Comparison with the baseline methods. The results of **CES** evaluation and its pruned version (denoted hereinafter **CES[50]** or **CES[100]** according to the number of top sentences that are chosen) are summarized in Table 1. Comparing the unpruned version of **CES** side-by-side with the baselines, we can observe that, **CES** outperforms all baselines on ROUGE-2 and ROUGE-SU4. On ROUGE-1, **CES** is superior in most cases, except when compared

[4] ROUGE-1.5.5.pl -a -c 95 -m -n 2 -2 4 -u -p 0.5 -l 250

(a) Recall:	**DUC 2005**			**DUC 2006**			**DUC 2007**		
Method	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4
BI-PLSA	0.3602	0.0676	-	0.3938	0.0849	-	-	-	-
HierSum	-	-	-	0.4010	0.0860	0.1430	0.4240	0.1180	0.1670
SpOpt-Δ	-	-	-	0.3996	0.0868	0.1422	0.4236	0.1110	0.1647
QODE	0.3751	0.0775	0.1341	0.4015	0.0928	0.1479	0.4295	0.1163	0.1685
CES	0.4012 (0.4004-0.4019)	0.0788 (0.0784-0.0791)	0.1388 (0.1384-0.1391)	0.4297 (0.4291-0.4302)	0.0973 (0.0970-0.0976)	0.1554 (0.1551-0.1557)	0.4495 (0.4492-0.4497)	0.1178 (0.1176-0.1180)	0.1729 (0.1727-0.1730)
CES[50]	0.4035 (0.4033-0.4038)	0.0794 (0.0793-0.0795)	0.01391 (0.1389-0.1392)	0.4301 (0.4299-0.4303)	0.0969 (0.0968-0.0971)	0.1565 (0.1564-0.1567)	0.4545 (0.4542-0.4547)	0.1202 (0.1201-0.1203)	0.1754 (0.1753-0.1755)
CES[100]	0.4033 (0.4030-0.4036)	0.0794 (0.0792-0.0796)	0.1389 (0.1387-0.1390)	0.4300 (0.4299-0.4301)	0.0969 (0.0968-0.0970)	0.1563 (0.1563-0.1564)	0.4543 (0.4542-0.4544)	0.1202 (0.1201-0.1203)	0.1750 (0.1750-0.1751)

(b) F-Measure:	**DUC2005**			**DUC2006**			**DUC2007**		
Method	R-1	R-2	R-SU4	R-1	R-2	R-SU4	R-1	R-2	R-SU4
MultiMR	0.3690	0.0683	-	0.4030	0.0850	-	0.4204	0.1030	-
DocRebuild[BoW]	-	-	-	0.3863	0.0723	0.1301	0.4115	0.0923	0.1483
DocRebuild[PV]	-	-	-	0.4086	0.0848	0.1445	0.4272	0.1030	0.1581
CTSUM	-	-	-	-	-	-	0.4266	0.1083	0.1616
CES	0.3757 (0.3750-0.3764)	0.0739 (0.0735-0.0742)	0.1300 (0.1296-0.1304)	0.4044 (0.4039-0.4049)	0.0916 (0.0913-0.0919)	0.1463 (0.1460-0.0.1465)	0.4241 (0.4238-0.4243)	0.1111 (0.1109-0.1113)	0.1631 (0.1629-0.1632)
CES[50]	0.3778 (0.3776-0.3781)	0.0745 (0.0744-0.0746)	0.1302 (0.1300-0.1303)	0.4047 (0.4045-0.4049)	0.0913 (0.0912-0.0914)	0.1473 (0.1472-0.1474)	0.4286 (0.4284-0.4288)	0.1134 (0.1133-0.1135)	0.1653 (0.1652-0.1654)
CES[100]	0.3776 (0.3773-0.3779)	0.0745 (0.3773-0.3779)	0.1300 (0.1299-0.1302)	0.4046 (0.4045-0.4047)	0.0913 (0.0912-0.0914)	0.1471 (0.1470-0.1472)	0.4284 (0.4283-0.4285)	0.1133 (0.1132-0.1134)	0.1650 (0.1649-0.1651)

Table 1: Results of ROUGE Recall and F-Measure evaluation on DUC 2005, 2006, and 2007 datasets. The CES values depicted in the table are the mean taken over 30 runs. Below each CES ROUGE result the 95% confidence interval is given. The compared results are the best reported values in the corresponding baselines. The symbol "–" represents an unknown result in case it is not reported by the corresponding baseline.

against the F-Measure ROUGE-1 of **DocRebuild**[PV] on DUC 2006-2007 benchmarks. However, as was mentioned above, the performance of **DocRebuild**[PV] might be influenced from the fact that word representations are learned on the input documents [6].

3.3.2 Effect of sentence pruning. As we can further observe, both pruned versions **CES[50]** and **CES[100]** further outperform the ROUGE-1 of **DocRebuild**[PV] on the DUC 2007 dataset. Comparing the unpruned and pruned versions of **CES** side-by-side, we can further observe that, the pruned versions provide comparable quality (even better sometimes) to that of the unpruned version. Moreover, the pruned version of **CES** is much more efficient. For example, using a single 8-core machine (16 threads in parallel) with 5GB heap memory, **CES** converged within 40.72 ± 0.55 iterations (or 44.58 ± 3.4 seconds) per topic statement on average. Compared to that, its **CES[50]** (pruned) version converged much faster within 25.73 ± 0.21 iterations (or 2.63 ± 0.03 seconds) per topic statement on average; i.e., about 17-times speedup in runtime. Overall, this illustrates that **CES** is both effective and efficient.

3.3.3 CES performance stability. Further analyzing **CES** average performance confidence intervals demonstrates that, even though it is possible that **CES** may generate different summaries between runs for the same topic statement, such variability is actually very small. This in turn, shows that, **CES** (stochastic) summarization policies are quite stable.

REFERENCES

[1] Hoa Trang Dang. Overview of duc 2005. In *Proceedings of the document understanding conference*, volume 2005, pages 1–12, 2005.

[2] Gareth E. Evans, Jonathan M. Keith, and Dirk P. Kroese. Parallel cross-entropy optimization. In *Proc. of WSC*, pages 2196–2202, 2007.

[3] Aria Haghighi and Lucy Vanderwende. Exploring content models for multi-document summarization. In *Proceedings of NAACL '09*.

[4] Zhanying He, Chun Chen, Jiajun Bu, Can Wang, Lijun Zhang, Deng Cai, and Xiaofei He. Document summarization based on data reconstruction. In *Proceedings of AAAI'12*.

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

[6] Shulei Ma, Zhi-Hong Deng, and Yunlun Yang. An unsupervised multi-document summarization framework based on neural document model. In *Proceedings of COLING 2016*.

[7] Ani Nenkova, Kathleen McKeown, et al. Automatic summarization. *Foundations and Trends® in Information Retrieval*, 5(2–3):103–233, 2011.

[8] Reuven Y Rubinstein and Dirk P Kroese. *The cross-entropy method: a unified approach to combinatorial optimization, Monte-Carlo simulation and machine learning*. Springer, 2004.

[9] Chao Shen, Tao Li, and Chris H. Q. Ding. Integrating clustering and multi-document summarization by bi-mixture probabilistic latent semantic analysis (plsa) with sentence bases. In *Proceedings of AAAI'11*.

[10] Xiaojun Wan and Jianguo Xiao. Graph-based multi-modality learning for topic-focused multi-document summarization. In *Proceedings of IJCAI'09*.

[11] Xiaojun Wan and Jianmin Zhang. Ctsum: Extracting more certain summaries for news articles. In *Proceedings of SIGIR '14*.

[12] Shengli Wu. *Data fusion in information retrieval*, volume 13. Springer Science & Business Media, 2012.

[13] Yang Xu, Gareth J.F. Jones, and Bin Wang. Query dependent pseudo-relevance feedback based on wikipedia. In *Proceedings of SIGIR '09*.

[14] Jin-ge Yao, Xiaojun Wan, and Jianguo Xiao. Compressive document summarization via sparse optimization. In *Proceedings of IJCAI'15*.

[15] Sheng-hua Zhong, Yan Liu, Bin Li, and Jing Long. Query-oriented unsupervised multi-document summarization via deep learning model. *Expert Syst. Appl.*, 42(21):8146–8155, November 2015.