Information Processing and Management 50 (2014) 443–461

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Text summarization using Wikipedia

Yogesh Sankarasubramaniam [a], Krishnan Ramanathan [a,*], Subhankar Ghosh [b,1]

[a] HP Labs India, Bangalore, India
[b] SAS Institute, San Diego, CA, United States

ARTICLE INFO

Article history:
Received 18 January 2013
Received in revised form 20 January 2014
Accepted 4 February 2014
Available online 6 March 2014

Keywords:
Summarization
Wikipedia
Sentence ranking
Personalization

ABSTRACT

Automatic text summarization has been an active field of research for many years. Several approaches have been proposed, ranging from simple position and word-frequency methods, to learning and graph based algorithms. The advent of human-generated knowledge bases like Wikipedia offer a further possibility in text summarization – they can be used to understand the input text in terms of salient concepts from the knowledge base. In this paper, we study a novel approach that leverages Wikipedia in conjunction with graph-based ranking. Our approach is to first construct a bipartite sentence–concept graph, and then rank the input sentences using iterative updates on this graph. We consider several models for the bipartite graph, and derive convergence properties under each model. Then, we take up personalized and query-focused summarization, where the sentence ranks additionally depend on user interests and queries, respectively. Finally, we present a Wikipedia-based multi-document summarization algorithm. An important feature of the proposed algorithms is that they enable real-time *incremental summarization* – users can first view an initial summary, and then request additional content if interested. We evaluate the performance of our proposed summarizer using the ROUGE metric, and the results show that leveraging Wikipedia can significantly improve summary quality. We also present results from a user study, which suggests that using incremental summarization can help in better understanding news articles.

1. Introduction

Text summarization has seen renewed interest recently. The reason for this is twofold: first, summarization can help cope with the information overload, and second, small form-factor devices are becoming increasingly popular. Internet access on the move often happens in an attention-deficit situation where the user is capable of assimilating lesser content. Hence, it becomes important to present only the most relevant information. For example, while reading a news article, the user might first want to look at a short summary (about 50–100 words), and then request the full article if interested.

Radev, Hovy, and McKeown (2002) define a document summary as "a text that is produced from one or more texts, that conveys important information in the original text(s), and that is no longer than half of the original text(s) and usually significantly less than that". In other words, a good summary: (1) is short and (2) preserves important information. There is a rich history of literature, dating back to the 1950s, that aims to achieve these two objectives. While the earliest efforts were restricted to simple position and word-frequency methods (perhaps due to the limited computation available at the time), more recent work

* Corresponding author. Tel.: +91 80 33829145; fax: +91 80 26129434.
E-mail addresses: yogesh@hp.com (Y. Sankarasubramaniam), krishnan_ramanathan@hp.com (K. Ramanathan), subhankar.ghosh@gmail.com (S. Ghosh).
[1] Work done while the author was with HP Labs India.

http://dx.doi.org/10.1016/j.ipm.2014.02.001
0306-4573/

has leveraged learning and graph-based algorithms for generating better summaries. This paper aims to study a further possibility: utilizing a human-generated knowledge base, like Wikipedia, in conjunction with graph-based summarization.

Wikipedia is perhaps the best example of collaborative knowledge creation and sharing, whereby the entire information is readily available to anyone–anywhere–anytime. The fact that it contains topics and entities of interest to humans makes it especially useful for summarization tasks. Moreover, Wikipedia is constantly updated, and provides the topic quality required for generating good summaries. Thus, Wikipdia can serve as the basis for understanding salient concepts from the input text, which can then be used to extract summary sentences.

The main contributions of this work are: (1) We cast the Wikipedia-based summarization problem into a general sentence–concept bipartite framework, and propose an iterative ranking algorithm for selecting summary sentences. (2) We provide precise mathematical definitions and analysis of the iterative ranking algorithm, and derive several convergence results. (3) We study generalizations of the basic bipartite setup, including directed/weighted edges, personalization and query focusing of summaries, and extensions to multi-document summarization. Furthermore, our algorithms provide *incremental summarization* in real-time, so that users can first view an initial summary, and if interested, can then request additional content. We also provide novel connections of our proposed algorithms with latent topic models and other known optimization problems.

The rest of this paper is organized as follows. We first review related summarization literature in Section 2, and state our research objective in Section 3. Then, in Section 4, we present the Wikipedia-based single document summarizer which is based on a novel iterative sentence–concept ranking. We provide precise mathematical definitions and convergence analysis under different scenarios, starting with undirected binary sentence–concept mappings in Section 4.2, followed by a generalization to weighted and directed mappings in Section 4.4, and then personalized and query-focused summarization in Section 4.5. Next, the Wikipedia-based multi-document summarizer is presented in Section 5. Experiments and performance evaluation are presented in Section 6, and the paper is concluded in Section 7.

2. Related work

A few recent efforts have leveraged Wikipedia for summarization tasks.[2] This includes our early work Ramanathan, Sankarasubramaniam, Mathur, and Gupta (2009), where summary sentences are selected based on Wikipedia concept frequency thresholds; the work of Ye, Chua, and Lu (2009) on summarizing definitions from Wikipedia pages; Wikipedia-based feature selection approaches (Bawakid & Oussalah, 2010; Gong, Qu, & Tian, 2010); Wikipedia-based sentence similarity approaches Miao and Li (2010); and the recent work of (Pourvali & Abadeh, 2012a, 2012b) which leverage Wikipedia to form multiple independent graphs, and then use graph importance and lexical cohesion features for summarization. However, one shortcoming of the above approaches is that they do not leverage graph-based ranking algorithms, which can capture the inter-sentence dependence and also utilize the entire graph structure for ranking sentences.

On the other hand, there exist several graph-based ranking algorithms that have independently been proposed for summarization tasks. Prominent among these are LexRank proposed by Erkan and Radev (2004), the iSpreadRank of Yeh, Ke, and Yang (2008), and the work of (Mihalcea et al., 2004; Mihalcea & Tarau, 2005). A particularly interesting approach is that of (Mihalcea & Tarau, 2005), who seek to rank sentences using well-known algorithms like HITS (Kleinberg, 1999) or PageRank Brin and Page (1998). Their evaluations show that graph-based methods can outperform baseline summarizers, and are in fact, competitive with the best supervised summarization algorithms. However, the document graph in their case is formed by connecting sentences based on word overlap, and they do not leverage knowledge bases like Wikipedia.

Thus, we see that there are two parallel lines of work: one that leverages Wikipedia, but does not utilize graph-based ranking algorithms; and the other that uses graph-based ranking algorithms, but without leveraging a knowledge base like Wikipedia. In this paper, we aim to bridge this gap by considering graph-based summarization in conjunction with Wikipedia. Moreover, our proposed ranking algorithms run directly on a bipartite graph, which is a departure from prior work which typically use a document graph. We also present several novel convergence results, and evaluations from experiments and user studies.

3. Research objective

Our aim in this work is to consider graph-based summarization in conjunction with Wikipedia. The specific research objectives are fourfold: (1) to present a unified sentence–concept bipartite framework for Wikipedia-based summarization, and propose a novel iterative ranking algorithm for summarization within this framework; (2) to provide mathematical formulation and analysis of the iterative ranking algorithm, and derive convergence properties; (3) to generalize the above framework and analysis to include directed/weighted edges, personalization and query-focusing of summaries, and multi-document summarization; and (4) to evaluate the performance of the proposed algorithms, experimentally using the ROUGE metric, and also directly by involving the user.

This paper also introduces the *incremental summarization* property (see Definitions 1 and 2 in Section 4), which is key to providing additional summary content in real-time. Results from our user study (see Section 6.4) suggest that this feature

[2] There is a long history of summarization literature, and the aim of this discussion is not to provide an exhaustive survey. We only seek to highlight the key ideas that help motivate our present work. The interested reader is referred to Das and Martins (2007) and Mani and Maybury (1999) for literature surveys.

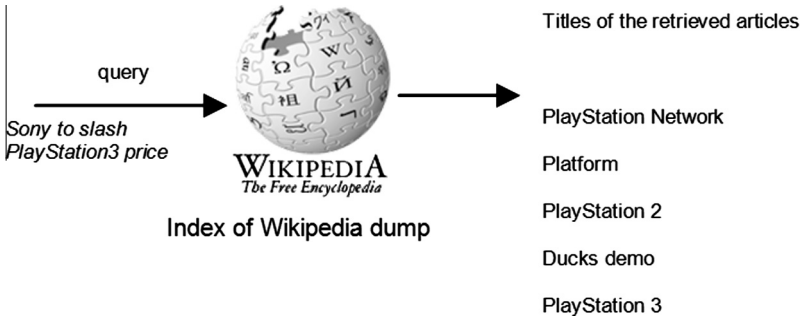

Fig. 1. Querying a Wikipedia index.

can help in better understanding news articles more than 70% of the time. We also present experimental evaluations in Section 6, using the ROUGE-1, ROUGE-2, precision and recall metrics, which show that leveraging Wikipedia can significantly improve summary quality.

Let us now start with Wikipedia-based single document summarization.

4. Single document summarization

The input here is typically a news article, book chapter, report, or any single piece of text, hereafter referred to as the input document *D*. The upshot of our approach is to first construct a bipartite sentence–concept graph, where the concepts represent Wikipedia article titles that are closest to the input sentences, and then rank the sentences for potential inclusion into a summary. We start by describing how to construct the bipartite sentence–concept graph in Section 4.1. Then, in Section 4.2, we describe the iterative ranking algorithm, and derive several convergence results. Thereafter, in Section 4.3, we discuss how to extract the summary sentences. Finally, Sections 4.4 and 4.5 provide details on the generalizations of the basic bipartite model: how to incorporate directed/weighted edges, and personalization and query-focusing of summaries.

4.1. Mapping sentences to Wikipedia concepts

Given an input document *D*, the sentences are first parsed and mapped to Wikipedia concepts (see for e.g. (Gabrilovich et al., 2006)). For this purpose, the entire Wikipedia corpus is indexed using the Lucene engine.[3] The sentences (after stopword removal and stemming) from *D* are then input as a query, and Wikipedia article titles are extracted from the results to the query ("hits" in Lucene terminology). These hits are precisely what are referred to as Wikipedia *concepts*.

This process is illustrated in Fig. 1 for an example sentence "Sony to slash PlayStation3 price". The top five Wikipedia concepts returned[4] are: *PlayStation Network*, *Platform*, *PlayStation 2*, *Ducks demo*, and *PlayStation 3*. It is worth noting that the concept *Platform* is rather generic, even though it has been returned as a top "hit" to the query. In Section 4.2, we will discuss the effect of such generic concepts, which in turn helps motivate our iterative sentence ranking algorithm.

Thus, using the procedure described above, we can obtain a sentence–concept mapping of *D*, i.e., we can map every sentence in *D* to their corresponding Wiki concepts. For a typical document (e.g. a news article or book chapter), we would expect to see some overlap of Wiki concepts across sentences, and this is indeed the case. It is thus useful to visualize the sentence–concept mapping as a bipartite graph, with one set of nodes representing the document sentences and the other set of nodes representing the Wiki concepts. An edge between a sentence node and a concept node indicates a mapping between the corresponding document sentence and Wiki concept, while the absence of an edge indicates that there is no mapping. Fig. 2 illustrates this for an example document of 3 sentences and 5 concepts. It is worth noting that concepts 2, 3 and 4 in this example map to multiple sentences.

4.2. Sentence ranking

The bipartite graph captures the relationship between sentences through the concept nodes. The question now is how to leverage this relationship in order to identify summary sentences. Before discussing our proposed method, let us first take a look at two natural approaches for this purpose, and see why they are not directly applicable here.

First, let us consider a "coverage" objective as in Filatova et al. (2004) and Takamura et al. (2009), where the aim is to select sentences that best cover a set of conceptual units. The authors of Filatova et al. (2004) provide several definitions of conceptual units in terms of the lexical or syntactic features extracted from the input text. However, the Wikipedia concept space considered in this paper is fundamentally different. It represents the topics and entities drawn from a

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

[4] In this example, the entire sentence is input as a query to the Wikipedia index. Alternatively, one could query using only the noun phrases or named entities – for e.g. "Sony", "PlayStation 3", and "price" in the above example.

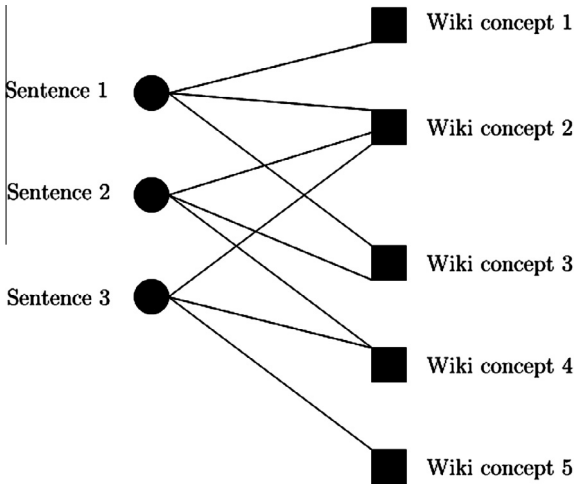

Fig. 2. Example of a sentence–concept bipartite graph.

human-generated knowledge base, and serves as a basis for understanding the input document. Thus, seeking to cover the entire basis may not yield a good summary. For example, we have found that news articles typically map to a large number of low-frequency concepts (concepts with node degrees 1 or 2 in the bipartite graph), and attempting to cover all of them would detract from the main story. Moreover, "coverage" objectives tend to treat high and low frequency concepts equally, which in turn affects the uniqueness of the summary.

Perhaps a more intuitive approach would be to select sentences that map to the most "important" concepts. However, it is not clear what this notion of "importance" means for Wiki concepts. One heuristic is to simply rank the concepts in decreasing order of their node degrees, i.e., their frequency of occurrence. In Fig. 2, for example, this would mean that Wiki concept 2 is ranked highest, followed by concepts 3 and 4. The heuristic then selects sentences mapping to the highest ranked concepts as the summary. We have found that this simple approach gives reasonable summaries (cf. our earlier work Ramanathan et al. (2009)).

However, it also has some serious limitations. Even after ranking the concepts according to their frequencies of occurrence, the problem of distinguishing between summary and non-summary sentences often persists. This is due to the fact that high-frequency concepts, by definition, map to several input sentences. To make this clear, let us go back to the example of Fig. 2, where all the three input sentences map to Wiki concept 2. Thus, we are still unable to select the summary sentences here. In practice, such an impasse occurs frequently with generic concepts like "Science", "Culture", and "Platform". The heuristic used in Ramanathan et al. (2009) to break the ties was to select sentences in their order of appearance, but clearly, this is not a robust solution. For instance, simply permuting the sentence order in the original text (without altering their content) would then drastically alter the summary.

Thus, there is a need to rank the sentences themselves, rather than just rank the Wikipedia concepts. We now describe one such ranking algorithm and study its properties. Specifically, we will see that the proposed ranking allows *incremental* summary generation.

Before proceeding further, let us introduce some notations and definitions. For ease of understanding, we use the example of Fig. 2 to relate to the notations. It must be remembered, however, that this example has been chosen only for illustration purposes, and is not intended to resemble sentence–concept mappings that are encountered in practice.

- Let $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ denote the n sentences in input D. For the example of Fig. 2, we have $n = 3$ and $\mathcal{S} = \{s_1, s_2, s_3\}$.
- Each sentence s_i is mapped to Wiki concepts as explained in Section 4.1. Let sentence s_i be mapped to a set $\mathcal{Z}_i$ of Wiki concepts. Then $\mathcal{Z} = \bigcup_{i=1}^{n} \mathcal{Z}_i$ represents the concept space, i.e., the entire set of concepts corresponding to all sentences in $\mathcal{S}$. Let $|\ \mathcal{Z}_i\ | = m_i$, and let $|\ \mathcal{Z}\ | = m$. For the example of Fig. 2, we have $m_i = 3$ for $i = 1, 2, 3$, and $m = 5$. In practice, we usually find $m \gg n$ (each sentence maps to several distinct concepts), and $m < \sum_{i=1}^{n} m_i$ (concepts overlap across sentences). For example, the news article New Planet yields $m = 527$ for $n = 35$ and $m_i = 20$.
- Write out the concept space $\mathcal{Z}$ and its subspace $\mathcal{Z}_i$ as follows:

$$\mathcal{Z} = \{z_1, z_2, \ldots, z_m\}, \tag{1}$$

where z_j represent the Wiki concepts, and

$$\mathcal{Z}_i = \{z_{i_1}, z_{i_2}, \ldots, z_{i_{m_i}}\}, \tag{2}$$

where each $z_{i_1}, z_{i_2}, \ldots, z_{i_{m_i}} \in \mathcal{Z}$. For example, from Fig. 2 we have $\mathcal{Z} = \{z_1, z_2, z_3, z_4, z_5\}$ with $\mathcal{Z}_1 = \{z_1, z_2, z_3\}$, $\mathcal{Z}_2 = \{z_2, z_3, z_4\}$, and $\mathcal{Z}_3 = \{z_2, z_4, z_5\}$.

- Analogously, represent the set of n_j sentences that map to Wiki concept z_j as:

$$\mathcal{S}_j = \{s_{j_1}, s_{j_2}, \ldots, s_{j_{n_j}}\}, \tag{3}$$

where each $s_{j_1}, s_{j_2}, \ldots, s_{j_{n_j}} \in \mathcal{S}$. For example, $\mathcal{S}_3 = \{s_1, s_2\}$ in Fig. 2.

- Denote the set $\{1, 2, \ldots, m\}$ by $\mathcal{M}$, and the set $\{1, 2, \ldots, n\}$ by $\mathcal{N}$. Further, denote the set of indices $\{i_1, i_2, \ldots, i_{m_i}\}$ in (2) by $\mathcal{M}_i$, and the set of indices $\{j_1, j_2, \ldots, j_{n_j}\}$ in (3) by $\mathcal{N}_j$.
- Let $\mathcal{U}_d = \{u_1, u_2, \ldots, u_d\}$ denote the set of d sentences comprising the summary. Our present scope is restricted to extractive summarization, where each $u_i \in \mathcal{S}$.

With these preliminaries, it is now natural to view the summarization problem as that of selecting the best $\mathcal{U}_d \subseteq \mathcal{S}$. However, we also require the following *incremental summarization* property:

Definition 1. Let $\mathcal{U}_d = \{s_i : i \in \mathcal{R}_d\}$, where $\mathcal{R}_d \subseteq \mathcal{N}$. A summarizer is said to satisfy the *incremental summarization* property if $\mathcal{R}_d \subset \mathcal{R}_{d+1}$ for each $d < n$, and the incremental index $r_{d+1} = \mathcal{R}_{d+1} \setminus \mathcal{R}_d$ is computable in constant time independent of d and n.

In other words, the incremental summarization property requires that longer summaries must be generated in real-time by simply adding sentences to shorter summaries. This requirement is necessary in our applications, where a short summary of a news article is first shown to the user, and after reading this summary, the user then decides whether to ask for additional content or not. In fact, the user can request for larger summaries in increments, all the way up to the entire article. More details on our experimental setup are provided in Section 6.4, along with results from a user study.

To support such repeated increments, it is critical to maintain the continuity of information sent to the user. For example, consider that the user first sees a 5-sentence summary, and then requests it to be increased to 7 sentences. Our solution ensures (by virtue of the incremental summarization property) that the existing 5 summary sentences remain unchanged, and only 2 new sentences are additionally shown. This allows the user to instantly recognize the additional information, thereby enabling successive increments until he/she is satisfied. On the other hand, recomputing a 7-sentence summary (without the incremental summarization property) could disrupt the information flow by presenting an entirely different set of sentences.

The above considerations prompt us to take an approach where sentence ranking is done only once, and summaries of any desired length are then generated using the sentence ranks. In the remainder of this section, we describe one such ranking algorithm[5] based on iterative updates, and study its convergence properties.

Let G denote the bipartite sentence–concept graph. Our goal now is to rank the sentence nodes $s_1, s_2, \ldots, s_n$ in decreasing order of "importance". Intuitively, the most important sentences map to the most important concepts, and vice versa. However, there is no apriori notion of "importance" among the Wiki concepts $z_1, z_2, \ldots, z_m$. Recall that the most frequently occurring z_j are not necessarily the most important, since they cannot distinguish between summary and non-summary sentences. Moreover, any global notion of importance derived from the Wikipedia graph is also not applicable here, since the summary depends only on the concepts $z_j \in \mathcal{Z}$ for the given document D. Thus, it appears that the importance of Wiki concepts is tied to the importance of the sentences themselves: one determines the other in a mutually reinforcing manner.

We thus proceed to rank sentences using iterative updates on G. Let us associate a score $x_i^{(k)}$ with sentence s_i, and a score $y_j^{(k)}$ with concept z_j, corresponding to iteration k. The updates for iteration $k+1$ are as follows:

$$\begin{aligned} y_j^{(k+1)} &= \sum_{i \in \mathcal{N}_j} x_i^{(k)}, \quad \forall j \in \mathcal{M}, \\ x_i^{(k+1)} &= \sum_{j \in \mathcal{M}_i} y_j^{(k)}, \quad \forall i \in \mathcal{N}, \end{aligned} \tag{4}$$

with the initialization $x_i^{(0)} = 1/\sqrt{n}$. The above updates are repeated K times, and the sentences are then ranked in decreasing order of scores $x_j^{(K)}$. The sentence-scores are prevented from growing without bound by normalizing them after each iteration, so that $\sum_{i \in \mathcal{N}} \left(x_i^{(k)}\right)^2 = 1$ for each k. The pseudocode now follows:

Algorithm 1. SENTENCE RANKER 1

Input: Sentence–concept mapping G
Output: Sentence/concept scores and ranks

1: Initialize $x_i^{(0)} = 1/\sqrt{n}, \quad j \in \mathcal{N}$
2: **for** $k = 0$ to $K - 1$ **do**
3: $\quad y_j^{(k+1)} = \sum_{i \in \mathcal{N}_j} x_i^{(k)}, \quad \forall j \in \mathcal{M}$
4: $\quad x_i^{(k+1)} = \sum_{j \in \mathcal{M}_i} y_j^{(k)}, \quad \forall i \in \mathcal{N}$
5: $\quad$ Normalize $x_i^{(k+1)} = \dfrac{x_i^{(k+1)}}{\sqrt{\sum_{i \in \mathcal{N}} \left(x_i^{(k+1)}\right)^2}}, \quad \forall i \in \mathcal{N}$
6: **end for**
7: Rank sentences in descending order of scores $\mathbf{r} = \arg\left(\text{descend}\left(x_1^{(K)} x_2^{(K)} \ldots x_n^{(K)}\right)\right)$

[5] We do not endeavor here to study all possible approaches that leverage the sentence–concept mapping and satisfy the *incremental summarization* property. Such a study needs to be conducted within an axiomatic framework, and is beyond the scope of this paper.

The number of iterations K can either be fixed, or can be arrived at using a stopping criterion, for e.g. $\sum_{i\in\mathcal{N}}\left(x_i^{(k+1)}-x_i^{(k)}\right)^2 \leqslant \epsilon$, for a suitably chosen ϵ. In practice, we have observed that the sentence-scores are steady within 5 to 10 iterations. In fact, the following result holds under a minor assumption:

Theorem 1. *The sentence score vector* $\mathbf{x}^{(k)} = \left(x_1^{(k)} x_2^{(k)} \ldots x_n^{(k)}\right)^T$ *converges to the principal eigenvector of* GG^T*, where* G *denotes the* $n \times m$ *biadjacency matrix corresponding to the sentence–concept mapping.*

Proof. The sentence–concept mapping described in Section 4.1 can be represented as an $n \times m$ binary matrix G as follows: define $g_{ij} = 1$ if the ith sentence is mapped to the jth concept, and $g_{ij} = 0$ otherwise.[6] G is called the biadjacency matrix of the sentence–concept mapping. For the example shown in Fig. 2, we have $n = 3, m = 5$, and $G = \begin{pmatrix} 1 & 1 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 \\ 0 & 1 & 0 & 1 & 1 \end{pmatrix}$. The rest of the proof relies on standard results from linear algebra, and is completed in Appendix A. □

As an example, let us apply Sentence Ranker 1 to the example of Fig. 2. At the end of the first iteration, we find that $\mathbf{x}^{(1)} = (0.54550.63640.5455)^T$; at the end of the second iteration $\mathbf{x}^{(2)} = (0.54210.64200.5421)^T$; and so on, soon converging to the principal eigenvector of GG^T, which is $(0.54180.64260.5418)^T$. Thus, Sentence Ranker 1 ranks Sentence 2 to be the highest.

The convergence of concept scores follows similarly:

Corollary 2. *The vector of concept scores* $\mathbf{y}^{(k)} = \left(y_1^{(k)} y_2^{(k)} \ldots y_m^{(k)}\right)^T$ *converges in the direction of the principal eigenvector of* G^TG.

Remark 1. Theorem 1 implies that the steady-state sentence score vector can be computed using any standard eigenvector algorithm. However, running the score updates for a fixed number (K) of iterations is computationally more efficient. Also, it serves to illustrate the sentence–concept coupling through the mutual reinforcements.

Remark 2. Sentence Ranker 1, in conjunction with the above convergence property, is reminiscent of the HITS (Kleinberg, 1999) algorithm for ranking search results. However, it is worth noting that our setup comes from an undirected bipartite sentence–concept graph, as opposed to the directed link graph used in HITS. Moreover, we neither seek nor find "hubs" and "authorities" as defined in HITS; and we provide several new results and generalizations (including Sections 4.4 and 4.5, Proposition 3 below, and Appendix B) that have significance for Wikipedia-based summarization.

The following result provides an intuitive connection between the iterative sentence–concept updates (4) and a known optimization problem. The proof is again deferred to Appendix A.

Proposition 3. *Let* $\mathbf{x}^\star$ *be the solution to the following optimization:*

$$\arg\max_{\|\mathbf{x}\|=1} \left(\|G^T\mathbf{x}\|\right)^2,$$

where $\|.\|$ *denotes the* L^2*-norm. Then* $\lim_{k\to\infty}\mathbf{x}^{(k)} = \mathbf{x}^\star$.

Proposition 3 can be seen as an interpretation of Sentence Ranker 1. It points out that the sentence scores converge to the solution of an optimization problem. Essentially, the sentence–concept mutual reinforcement (4) finds sentence scores that maximize the square-sum of concept scores.[7]

Finally, we make an interesting connection between Sentence Ranker 1 and a generative latent topic model. We show that under a bag-of-words assumption, the generative weights assigned to concepts follows the same relative ordering as the dominant eigenvector of G^TG. The details are given in Appendix B.

4.3. Extracting the summary

The output of Sentence Ranker 1 is the vector of ordered indices $\mathbf{r} = \arg\left(\text{descend}\left(x_1^{(K)} x_2^{(K)} \ldots x_n^{(K)}\right)\right)$. Let $\mathcal{R}_d = \{r_1, r_2, \ldots, r_d\}$ denote the set of d leading indices in $\mathbf{r}$. Then, the d-sentence summary is given by $\mathcal{U}_d = \{s_i : i \in \mathcal{R}_d\}$. Thus, we find that $\mathcal{R}_d \subset \mathcal{R}_{d+1}$ for each $d < n$, which is the first requirement for *incremental summarization*, given in Definition 1. The

[6] We have intentionally used G to denote both the bipartite graph and the biadjacency matrix corresponding to the sentence–concept mapping. These are two different visualizations of the same mapping, and are hence used interchangeably.

[7] In the language of linear algebra, $\mathbf{x}^\star$ is known as the *direction of maximum gain* of G^T.

second requirement is that the incremental index $r_{d+1} = \mathcal{R}_{d+1} \setminus \mathcal{R}_d$ must be computed in constant time independent of d and n. This is true under a reasonable assumption that $\mathbf{r}$ is precomputed.[8]

Thus, longer summaries can be generated *incrementally* by simply adding sentences to shorter summaries. However, in some cases, it is desirable to specify the summary-size in words instead of sentences: for example, default summaries are usually set to 50 or 100 words. Let $l(s_i)$ denote the length, in words, of sentence s_i. Analogous to Definition 1, we now define the *word-incremental summarization* property as follows:

Definition 2. Let $\mathcal{U}_d = \{s_i : i \in \mathcal{R}_d\}$ denote the d-word summary, where $\mathcal{R}_d \subseteq \mathcal{N}$. A summarizer is said to satisfy the *word-incremental summarization* property if $\mathcal{R}_d \subseteq \mathcal{R}_{d+\delta}$ for every $\delta > 0$, $d < \sum_{i=1}^{n} l(s_i) - \delta$; and the incremental indices $\mathcal{R}_{d+\delta} \setminus \mathcal{R}_d$ are computable in time independent of d and n.

Constructing the d-word summary $\mathcal{U}_d$ requires solving the following optimization problem:

$$\begin{aligned} \text{Maximize} \quad & \sum_{s_i \in \mathcal{U}_d} x_i^{(K)} \\ \text{such that} \quad & \sum_{s_i \in \mathcal{U}_d} l(s_i) \leqslant d \end{aligned}$$

This is equivalent to the *0–1 Knapsack problem* Cormen, Leiserson, Rivest, and Stein (2009), which can be solved exactly using dynamic programming. However, the resulting solution does not satisfy the word-incremental summarization property. Hence, we take recourse to the following greedy approximation[9] for constructing $\mathcal{U}_d$: first derive the per-word rank ordering $\mathbf{r} = \arg\left(\text{descend}\left(\frac{x_1^{(K)}}{l(s_1)}, \frac{x_2^{(K)}}{l(s_2)} \cdots \frac{x_n^{(K)}}{l(s_n)}\right)\right)$; then construct the d-word summary as $\mathcal{U}_d = \{s_i : i \in \mathcal{R}_{\tilde{d}}\}$, where $\tilde{d} = \arg\min_j \left|\sum_{i \in \mathcal{R}_j} l(s_i) - d\right|$. It can be verified that $\mathcal{R}_d \subseteq \mathcal{R}_{d+\delta}$ for every $\delta > 0$, $d < \sum_{i=1}^{n} l(s_i) - \delta$; and that the incremental indices are computable in real time under the assumption that the per-word rank ordering is precomputed. Thus, word-incremental summaries can now be generated in real time by simply selecting the next few sentences (from the per-word rank ordering) satisfying the length constraint.

4.4. Incorporating concept significance

Thus far, we have assumed that the sentence–concept mappings are binary, i.e., $g_{ij} \in \{0, 1\}$. This simple case serves to set up and analyze Sentence Ranker 1, but has some shortcomings. For example, it does not capture the relevance of a Wikipedia concept for a given sentence, and hence cannot distinguish between a generic introductory sentence, and one that provides specific information. For e.g., a sentence that simply mentions the word "planet", and another sentence that actually talks about news of a planet discovery, are both mapped to the same Wiki concept "planet". In such cases, it is important to distinguish the strength of Wiki concept mappings using real-valued g_{ij}, as opposed to binary 0/1 values. For example, g_{ij} could represent the estimates of conditional probabilities $P(z_j \mid s_i)$.

The question now is how to derive the mappings $P(z_j \mid s_i)$. A popular solution in literature is to use generative models. For instance, El-Arini, Veda, Shahaf, and Guestrin (2009) face a similar problem while assessing feature significance of a blog post, and the authors propose to use an LDA model learned on the noun phrases and named entities extracted from a blog corpus. However, the Wikipedia concept space $\mathcal{Z}$ in our case is actually explicit, and not latent. Thus, we can derive the conditional probabilities using the same method of Section 4.1, by measuring the strength of each "hit". Here, we assume that the Wikipedia articles are stable, and that their length is representative of an adequate description of the topic. Thus, we do not normalize the hit score by the length of the Wikipedia article. When querying is done using the entire sentence, the returned Lucene weights can directly act as estimates for $P(z_j \mid s_i)$. However, if the extracted noun phrases or named entities are used for querying, then the returned Lucene weights must be appropriately combined. Our implementation takes the former approach, but either way, the entries g_{ij} can now be set to the estimated $P(z_j \mid s_i)$.

With this approach, the sentence–concept mappings are no longer undirected. The probabilities $P(z_j \mid s_i)$ represent the entries of the *forward* mapping G, i.e., from sentences to concepts. The corresponding *backward* mapping H has entries $h_{ij} = P(s_i \mid z_j)$ representing the return path from concepts to sentences. With the forward (G) and backward (H) matrices, the iterative updates take the following form:

$$\begin{aligned} y_j^{(k+1)} &= \sum_{i \in \mathcal{N}} g_{ij} x_i^{(k)}, \quad \forall j \in \mathcal{M} \\ x_i^{(k+1)} &= \sum_{j \in \mathcal{M}} h_{ij} y_j^{(k)}, \quad \forall i \in \mathcal{N}, \end{aligned} \tag{5}$$

with the initialization $x_i^{(0)} = 1/\sqrt{n}$ as before.

[8] This can be achieved in practice by running Sentence Ranker 1 offline, once for each input document or news article of interest. It is worth noting that the bipartite sentence–concept model admits a natural parallelization.

[9] It is well-known that the greedy algorithm proposed here does not provide any performance guarantees for the 0–1 Knapsack problem. However, it works well in practice since the variation in $l(s_i)$ is small for most text documents.

The h_{ij} can be estimated using Bayes' rule: $P(s_i \mid z_j) = \frac{P(z_j|s_i)P(s_i)}{P(z_j)}$, where $P(s_i)$ and $P(z_j)$ denote the *apriori* sentence and concept probabilities, respectively. If the sentences and concepts are assumed to be apriori indistinguishable (i.e., they are equally likely to contribute towards the summary), then $P(s_i) = 1/n$ and $P(z_j) = 1/m$, whereby[10] $P(s_i \mid z_j) = \frac{m}{n}P(z_j \mid s_i)$ and $H = \frac{m}{n}G$. In this case, the backward matrix is just a scaled version of the forward matrix, and the rest of the analysis and results of Theorem 1 and Proposition 3 carry over.

However, the apriori probabilities may not always be equal. For e.g., the concept probabilities $P(z_j)$ could be biased based on user preferences, queries or trends. In such cases, H is no longer a scaled version of G, and the iterations (5) evolve as:

$$\begin{aligned}\mathbf{y}^{(k+1)} &= G^T\mathbf{x}^{(k)},\\ \mathbf{x}^{(k+1)} &= \frac{H\mathbf{y}^{(k+1)}}{\|H\mathbf{y}^{(k+1)}\|},\end{aligned} \tag{6}$$

where the sentence scores are normalized, as before, after each iteration. Thus, $\mathbf{y}^{(k+1)} = (G^TH)^kG^T\mathbf{x}^{(0)}$, and $\mathbf{x}^{(k+1)}$ is the unit vector in the direction of $(HG^T)^k\mathbf{x}^{(0)}$. By setting $A = HG^T$, we note that

$$\begin{aligned}a_{ij} &= \sum_{z_l\in\mathcal{Z}} P(s_i \mid z_l)P(z_l \mid s_j),\\ &= P(s_i)\sum_{z_l\in\mathcal{Z}} \frac{P(z_l \mid s_i)P(z_l \mid s_j)}{P(z_l)},\end{aligned} \tag{7}$$

and similarly

$$a_{ji} = P(s_j)\sum_{z_l\in\mathcal{Z}} \frac{P(z_l \mid s_j)P(z_l \mid s_i)}{P(z_l)}. \tag{8}$$

Thus, we find that A is symmetric (i.e., $a_{ij} = a_{ji}$) if $P(s_i)$ is equal for all i. That is, the matrix HG^T is symmetric under the assumption that the sentences are apriori indistinguishable, regardless of the apriori concept probabilities ($P(z_l)$ values). In such a case, Theorem 1 and Proposition 3 hold in the following modified forms. We skip the proofs as they are similar.

Theorem 4. *Let G and H denote the $n \times m$ matrices corresponding to the forward sentence-to-concept and backward concept-to-sentence mappings, respectively, and let $A = HG^T$. If $P(s_1) = P(s_2) = \cdots = P(s_n)$, then the sentence score vector $\mathbf{x}^{(k)} = \left(x_1^{(k)}x_2^{(k)}\ldots x_n^{(k)}\right)^T$ converges to the principal eigenvector of A.*

The following result provides an intuitive connection between the forward–backward iterations (5) and a known optimization problem. Essentially, we find the sentence scores converge to the maximizing direction of the quadratic form of A.

Proposition 5. *Let $\mathbf{x}^\star$ be the solution to the following quadratic optimization problem:*

$$\arg\max_{\|\mathbf{x}\|=1} \mathbf{x}^TA\mathbf{x}.$$

Then $\lim_{k\to\infty}\mathbf{x}^{(k)} = \mathbf{x}^\star$.

Remark 3. Theorem 4 and Proposition 5 hold even when the entries h_{ij} and g_{ij} do not actually represent the conditional probabilities $P(z_j \mid s_i)$ and $P(s_i \mid z_j)$, respectively. They could instead simply represent non-negative scores for the sentence-to-concept and concept-to-sentence mappings. The only requirement is that $A = HG^T$ be symmetric. In particular, it is not required here that G^TH be symmetric.

Finally, we consider the case when $P(s_i)$ could also be biased. This is relevant in news summarization, for example, where the first few sentences typically lead with the headlines and the main story. In such cases, the $P(s_i)$ values may be biased according to the position of sentence s_i. The iterative updates evolve as before in (6), but the matrix $A = HG^T$ is no longer symmetric, and Theorem 4 does not hold. However, since A is non-negative, the sentence scores can be shown to still converge under a minor restriction. We require the following definition in order to proceed:

Definition 3. Let A be an $n \times n$ matrix with non-negative entries a_{ij}. Then A is said to be *primitive* if, for some positive integer n_0, the matrix A^{n_0} is positive, i.e., the entries $a_{ij}^{n_0} > 0$.

We now state the following results without proof. The proofs rely on Perron–Frobenius Theory, and the interested reader is referred to Seneta (1981) for the details.

[10] Note that given $P(z_j \mid s_i)$ and $P(s_i)$, $P(z_j)$ is determined as $P(z_j) = \sum_{s_i\in\mathcal{S}} P(z_j \mid s_i)P(s_i)$. The indistinguishability assumption here must be understood in light of the apriori $P(z_j)$ being such that this relation holds.

Theorem 6. *Let G and H denote the $n \times m$ matrices corresponding to the forward sentence-to-concept and backward concept-to-sentence mappings, respectively, and let $A = HG^T$. If A is primitive, then the sentence score vector $\mathbf{x}^{(k)} = \left(x_1^{(k)} x_2^{(k)} \dots x_n^{(k)}\right)^T$ converges to the principal eigenvector of A.*

The principal eigenvector of a primitive matrix A is referred to as its Perron–Frobenius (PF) eigenvector. The following result provides a connection between the steady-state sentence score vector and the optimal point of a certain max–min optimization, through the PF eigenvector. Essentially, if A is primitive, then sentence scores from (5) converge in the direction maximizing the minimum *growth factor* over components.

Proposition 7. *Let $\mathbf{x}^\star$ be the solution to the following optimization:*

$$\arg\max_{\mathbf{x}>\mathbf{0}} \min_i \frac{(Ax)_i}{x_i},$$

where x_i denotes the ith component of a vector $\mathbf{x} = (x_1 x_2 \dots x_n)^T$. Then $\lim_{k\to\infty} \mathbf{x}^{(k)} = \mathbf{x}^\star$.

The comments of Remark 3 are also applicable here, with the only change that $A = HG^T$ must now be primitive, instead of symmetric. One might ask what happens if A is neither primitive nor symmetric. Given that A is non-negative, and that the initialization $x_i^{(0)} = 1/\sqrt{n}$ is symmetric and positive, further analysis is possible. However, that goes beyond our present scope, and we close by saying that Sentence Ranker 1 yields steady-state sentence scores for practically all sentence–concept mappings.

4.5. Personalized and query-focused summaries

In our analysis thus far, we assumed that the sole input to the summarizer is a document or text content. However, one might want to provide additional inputs to bias the summary in certain ways: for e.g., to personalize the summary, or to present it in response to queries. In each case, there is an explicit notion of concept importance obtained through the user interests or queries, respectively. This is in contrast with our earlier setup, where concept importance was implicit and tied to the summarization itself.

Let $\Pi = \{\pi_1, \pi_2, \dots, \pi_t\}$ denote the set of t user interests, with corresponding weight vector $\mathbf{w} = (w_1 w_2 \dots w_t)^T$, i.e., interest π_l carries weight w_l. The case of user queries is similar, with π_l now denoting the lth query. In the rest of this section, we only study the personalization aspect. Query-focused summaries can be derived along similar lines.

To fix ideas, let us consider the simplest case when $\Pi = \mathcal{Z}$, i.e., the set of user interests is specified on the Wiki concept space $\mathcal{Z}$. This is hardly ever the case in practice, but serves to illustrate the approach. As before, we will require the summarizer to satisfy the incremental summarization property (Definition 1 at the sentence-level and Definition 2 at the word-level). We thus seek to rank the sentences once, and then extract summaries of any desired length. The difference in this case, however, is that the concept distribution is already specified through $\mathbf{w}$. This means that the vector $\mathbf{x}$ of sentence scores is simply the solution to a system of linear equations $\mathbf{w} = G^T\mathbf{x}$, where G denotes the forward sentence-to-concept matrix as defined in Section 4.4. For the typical case of $m > n$, the system of equations is overdetermined, and let $\mathbf{x}^\star = \left(x_1^\star x_2^\star \dots x_n^\star\right)^T$ denote the corresponding least squares solution Strang (1988). Note that there are no iterative updates here, and we do not require $\mathbf{x}^\star$ or $\mathbf{w}$ to have unit norm. Also G need not be stochastic, though we make the usual assumption that the g_{ij} are non-negative.

Let us now turn to the more realistic scenario, where Π is different from $\mathcal{Z}$. Our approach is to map the interests π_l to concepts $z_j \in \mathcal{Z}$. We use the Lucene Wiki index as before, but now additionally collect "hits" where π_l and z_j co-occur. This allows us to estimate the joint distribution $P(z_j, \pi_l)$. Our aim now is to derive the conditional probabilities $P(\pi_l \mid s_i)$, and then find the least squares solution $\mathbf{x}^\star$ to the equations $\mathbf{w} = B^T\mathbf{x}$, where the entries of matrix B are $b_{il} = P(\pi_l \mid s_i)$. To this end, we make the standard modeling assumption that $\mathcal{S}$ and Π are related through $\mathcal{Z}$, i.e.,

$$P(\pi_l \mid s_i) = \sum_{z_j \in \mathcal{Z}} P(z_j \mid s_i) P(\pi_l \mid z_j), \tag{9}$$

where $P(z_j \mid s_i)$ are obtained as described in Section 4.4, and $P(\pi_l \mid z_j)$ can be estimated from the joint distribution as $P(\pi_l \mid z_j) = \frac{P(z_j,\pi_l)}{\sum_{\pi_l \in \Pi} P(z_j,\pi_l)}$. Once the solution $\mathbf{x}^\star$ is determined, the sentence ranks are given by $\mathbf{r} = \arg\left(\text{descend}\left(x_1^\star x_2^\star \dots x_n^\star\right)\right)$, and the personalized d-sentence or d-word summary can be formed as described in Section 4.3.

5. Multi-document summarization

We now turn to multi-document summarization, where the input is typically a cluster of related documents, for e.g. news articles pertaining a single story, or the top hits from a search result. In such cases, the inputs can have similar or even identical sentences, and summarization involves not only ranking sentences but also filtering out redundancy. Clearly, Sentence Ranker 1 cannot be used in its present form, since it would rank similar sentences close to each other. This can be problematic while selecting the summary sentences. For instance, while summarizing news articles pertaining to a single tory, the highest-ranked sentences could come from different news portals, and hence could be very similar, but they would all be

selected in any d-sentence summary, $d \geqslant 3$. Thus, we now proceed to formulate the multi-document summarization problem, and propose a revised sentence ranker that addresses this issue.

Let $\mathcal{S} = \{s_1, s_2, \ldots, s_n\}$ denote the multiset of n sentences derived from the multiple input documents. All other notations carry over from Section 4.2, but must be taken now with reference to the multiset $\mathcal{S}$. Our goal is to derive the multi-document summary $\mathcal{U}_d = \{u_1, u_2, \ldots, u_d\}$ consisting of d sentences chosen from $\mathcal{S}$. To ensure sufficient diversity, pairs of sentences u_i, u_j must not have significant overlap. Also, the incremental summarization property is required as before (see Definition 1).

Before describing our approach, let us briefly review some of the existing multi-document summarization methods. Perhaps the most natural approach would be to cluster the sentences in $\mathcal{S}$, and then select one representative sentence from each cluster. This is broadly the method followed in McKeown, Klavans, Hatzivassiloglou, Barzilay, and Eskin (1999) and Radev et al. (2002). Radev, Jing, Stys', and Tam (2004) Another popular approach is graph-based, where the sentences form the vertices of a graph, and edge weights indicate pairwise sentence similarity. Summary sentences are then selected based on suitable objective functions (Erkan & Radev, 2004; Mihalcea et al., 2004; Lin, Bilmes, & Xie, 2009; Lin et al., 2010). In some cases, the objective functions turn out to be submodular, for e.g. in Lin et al. (2009, 2010), whereupon standard approximations (Khuller, Mossb, & Naor, 1999; Krause et al., 2005) with known guarantees can be applied. In other cases, more complex algorithms have been proposed, for e.g. in McDonald (2007), where summary extraction is formulated as an integer linear program (ILP). There are also many other approaches which we are unable to elaborate here, and we refer the interested reader to Das and Martins (2007) for a detailed survey.

While the above methods are effective in their own right, they operate in a setup quite different from ours. None of them incorporate Wikipedia concept mappings as considered in this paper. Moreover, we have imposed the *incremental summarization* requirement, in light of which we do not formulate summarization as an optimal subset selection problem. Furthermore, for reasons mentioned earlier (see Section 4.2), the notion of sentence importance in our case is coupled to concept importance, thus resulting in a mutual reinforcement quite different from the above-mentioned approaches.

With this in mind, we take the following two-step approach[11] towards multi-document summarization: first, determine the steady-state sentence and concept scores using iterative updates similar to (5); next, derive sentence ranks based on a diversification objective. The latter step leverages the sentence and concept scores computed in the first step.

For the first step, the sentence–concept mapping and forward/backward matrices can be derived as in Section 4, but with the multiset $\mathcal{S}$ now as input. Let $\mathbf{x}^{(K)} = \left(x_1^{(K)} x_2^{(K)} \ldots x_n^{(K)}\right)^T$ and $\mathbf{y}^{(K)} = \left(y_1^{(K)} y_2^{(K)} \ldots y_m^{(K)}\right)^T$ denote the steady-state sentence and concept scores, respectively, at the end of iteration K. In the second step, we seek to rank the sentences based on their individual scores $x_i^{(K)}$, and also their pairwise similarity with other sentences. An analogous problem in literature is that of diversifying search results (see for e.g. (Agrawal, Gollapudi, Halverson, & Ieong, 2009; Clarke et al., 2008; Zhai, Cohen, & Lafferty, 2003; Skoutas, Minack, & Nejdl, 2010)). We follow a similar greedy ranking approach, though our objective function is specific to summarization. We thus have the Sentence Ranker 2 as follows:

Algorithm 2. Sentence Ranker 2

Input: Sentence–concept mapping G
Output: Sentence ranks

Initialize $x_i^{(0)} = 1/\sqrt{n}, \quad j \in \mathcal{N}$
for $k = 0$ to $K - 1$ **do**
  $y_j^{(k+1)} = \sum_{i \in \mathcal{N}} g_{ij} x_i^{(k)}, \quad \forall j \in \mathcal{M}$
  $x_i^{(k+1)} = \sum_{j \in \mathcal{M}} h_{ij} y_j^{(k)}, \quad \forall i \in \mathcal{N}$
  Normalize $x_i^{(k+1)} = \dfrac{x_i^{(k+1)}}{\sqrt{\sum_{i \in \mathcal{N}} \left(x_i^{(k+1)}\right)^2}}, \quad \forall i \in \mathcal{N}$
end for
Initialize $\mathbf{r} = r_1$, where $r_1 = \arg\max_i x_i^{(K)}$
for $j = 2$ to n **do**
  Update $\mathbf{r} \leftarrow (\mathbf{r} r_j)$,
  where $r_j = \arg\max_{i \in \mathcal{N} \setminus \mathcal{R}_{j-1}} c(\mathcal{R}_{j-1} \cup \{i\}) - c(\mathcal{R}_{j-1})$,
  $\mathcal{R}_{j-1} = \{r_1, r_2, \ldots, r_{j-1}\}$,
  and $c(\mathcal{L}) \triangleq \sum_{l \in \mathcal{L}} x_l^{(K)} - \alpha \sum_{u, v \in \mathcal{L}, u < v} Red(s_u, s_v)$
end for

[11] Our present scope does not permit a full axiomatic study of multi-document summarizers that leverage sentence–concept mappings and satisfy the *incremental summarization* property. Here, we only propose one algorithm that performs well in practice.

Here, $Red(s_u, s_v)$ denotes the redundancy between sentences s_u and s_v, and α is a suitable weighting factor. The redundancy $Red(s_u, s_v)$ depends on $\mathbf{y}^{(K)}$, and is computed as follows: let $\mathbf{y}^{(K)}|_u$ denote the concept score vector restricted to sentence s_u, i.e.,

$$y_j^{(K)}|_u = \begin{cases} y_j^{(K)} & \text{if } j \in \mathcal{M}_u \\ 0 & \text{otherwise,} \end{cases} \tag{10}$$

where $\mathcal{M}_u$ denotes the set of indices corresponding to concepts in $\mathcal{Z}_u$ (see (2)). In other words, $\mathbf{y}^{(K)}|_u$ is derived from $\mathbf{y}^{(K)}$ by setting to 0 the scores for those concepts that do not map to sentence s_u. Similarly, let $\mathbf{y}^{(K)}|_v$ denote the concept score vector restricted to sentence s_v. Then, $Red(s_u, s_v)$ is given by the cosine similarity between vectors $\mathbf{y}^{(K)}|_u$ and $\mathbf{y}^{(K)}|_v$.

A cost function $c : 2^{\mathcal{N}} \to \mathbb{R}$ is then defined as follows:

$$c(\mathcal{L}) \triangleq \sum_{l \in \mathcal{L}} x_l^{(K)} - \alpha \sum_{\substack{u,\, v \in \mathcal{L} \\ u < v}} Red(s_u, s_v) \tag{11}$$

where $\mathcal{L} \subseteq \mathcal{N}$ is any subset of the sentence indices. Thus, the cost associated with a subset of sentences is a (weighted) difference between the total score of sentences in that subset, and their total pairwise redundancy. The negative sign in (11) has the effect of penalizing redundancy, i.e., it favors selection of dissimilar sentences. Taken in conjunction with (10), this means that concept overlaps across sentences are penalized in proportion to their concept scores.

Thus, the sentences in multiset $\mathcal{S}$ can be rank-ordered using Sentence Ranker 2. Any d-sentence summary can then be formed by selecting the d leading sentences. On the other hand, to form a d-word summary, we require the per-word rank ordering. This can be obtained by slightly modifying steps 7 and 9 of Sentence Ranker 2 as follows:

$$\begin{aligned} r_1 &= \arg\max_i \frac{x_i^{(K)}}{l(s_i)}, \\ r_j &= \arg\max_{i \in \mathcal{N} \setminus \mathcal{R}_{j-1}} \frac{c(\mathcal{R}_{j-1} \cup \{i\}) - c(\mathcal{R}_{j-1})}{l(s_i)}, \quad j > 1, \end{aligned} \tag{12}$$

with the rest of the steps remaining unchanged. The d-word summary is then formed as described in Section 4.3. It is worth noting here that the *incremental summarization* property – Definition 1 at the sentence level, and Definition 2 at the word level – is satisfied by the proposed multi-document summarizer.

Summary coherence can be further imposed, if desired, by following time order and text order (sentences from oldest documents appear first) on the selected sentences. One such method is proposed by Barzilay, Elhadad, and McKeown (2001), where a combination of chronological and coherence constraints is used to order sentences.

6. Performance evaluation

In this section, we present experimental results for evaluating the performance of our Wikipedia-based summarizer. We also report the learnings from a user study conducted to assess the impact of incremental summarization. The upshot of our results is that leveraging Wikipedia can significantly improve summary quality. The Wikipedia-based summarizer not only improves upon baseline ROUGE scores, but also comes close to the optimum performance for news articles. Moreover, the use of incremental summarization can help in better understanding news articles.

Let us start with some details on the data set and metrics used in our evaluations.

6.1. Data set

The National Institute of Standards and Technology (NIST) has organized a yearly series of conferences called the Document Understanding Conference[12] (DUC) since 2001 DUC series. We use the DUC 2002 data, which consists of 567 English news articles DUC (2002). Single-document summarization was the first out of three tasks in DUC 2002, with the aim of automatically generating summaries of length 100 words or less. As a preprocessing step, stop-word removal and stemming was performed on each of the documents in the data set.

Since 2002, the single document summarization task has been dropped by DUC. However, the 2002 data set is available, and has been used frequently for evaluating new summarization algorithms. Another reason for our choice of this data set is that it is focused on news articles, which is our primary use case involving incremental summarization.

6.2. Evaluation metrics

We compute recall scores for 100-word summaries using the ROUGE metric Lin et al. (2003), which has been adopted by DUC for automatic summary evaluation. ROUGE relies on counting the number of overlapping word sequences between the candidate and reference summaries, and has been found to agree surprisingly well with human judgment Lin (2004). More

[12] Replaced by TAC (Text Analysis Conference) from 2008 onwards.

precisely, ROUGE-*N* is an *N*-gram recall measure computed between a candidate summary and a set of one or more reference summaries as follows:

$$\text{ROUGE} - N = \frac{\sum_{S \in \text{reference summaries}} \sum_{N\text{-gram} \in S} \text{Count}_{\text{match}}(N\text{-gram})}{\sum_{S \in \text{reference summaries}} \sum_{N\text{-gram} \in S} \text{Count}(N\text{-gram})} \quad (13)$$

where $\text{Count}_{\text{match}}(.)$ represents the maximum number of *N*-grams co-occurring in a candidate summary and the set of reference summaries, and Count (.) represents the number of *N*-grams in the set of reference summaries.

In our experiments, we first used the ROUGE toolkit ROUGE to compute the ROUGE-1 (unigram) and ROUGE-2 (bigram) scores, which have been found to agree most with human evaluation Lin et al. (2003). For this purpose, we first obtained the 100-word summaries of each of the 567 news articles using our Wikipedia-based single document summarizer. For comparison, we then computed the 100-word summaries using the following three techniques:

- *Leading sentences*: first few sentences are chosen the summary.
- *Random sentences*: summary sentences are chosen uniformly at random.
- *Best sentences*: summary sentences chosen manually to best match the reference summaries.

The first two are intended to serve as baselines, while the third provides a ceiling on the performance of any extractive summarization algorithm.

For computing the ROUGE scores, the reference summaries for each article were directly obtained from the DUC 2002 data set. In order to ensure a fair comparison, we further used the "-l 100" option in ROUGE to truncate longer summaries.

Next, in our second set of experiments, we computed the precision, recall and F-measure for summarizers that showed similar ROUGE-1 performance. These metrics, along with the corresponding ROUGE-2 scores, helped in differentiating the summarizers. We now present the detailed results.

6.3. Experimental results

Table 1 shows the ROUGE-1 and ROUGE-2 scores of our Wikipedia-based single document summarizer along with *Leading sentences*, *Random sentences*, *Best sentences*, and the five top-performing summarizers from DUC 2002 (Wan, Yang, & Xiao, 2006) – these systems are: System 28 (Schlesinger et al., 2002), System 21 (van Halteren, 2002), System 31 (Brunn, Chali, & Dufour, 2002), System 29 (Angheluta, De Busser, & Moens (2002)), and System 27 (Hirao, Sasaki, Isozaki, & Maeda, 2002). For further comparison, we also evaluated the ROUGE scores using the built-in AutoSummarize feature in MS Word.

As expected, we find that *Best sentences* shows the highest ROUGE-1 and ROUGE-2 scores of 0.51 and 0.29, respectively. The summaries here were extracted manually, with the aim of matching the reference summaries as much as possible. Thus, the ROUGE scores of *Best sentences* serve as an upper bound on the performance of any extractive summarizer for the DUC 2002 data set. It is interesting to note that the top-performing DUC 2002 summarizer comes within 0.03 of this upper bound for ROUGE-1. However, the difference in ROUGE-2 scores is greater.

At the other end, *Random sentences* shows the poorest ROUGE scores of 0.41 and 0.15. *Leading sentences* shows surprisingly high scores at first sight. In fact, the ROUGE-1 score is only 12% short of the *Best sentences* upper bound, and the ROUGE-2 score is as high as 75% of the upper bound. This suggests that simply selecting the first few sentences is a reasonable summarization strategy for news articles. Indeed, this is not surprising, given that most news articles lead with a gist of the story. Another reason could be that the evaluation is done against human-generated summaries, which naturally favors the leading lines in a news article.

In general, however, *Leading sentences* is far from optimal. A simple example would be to randomly reorder the sentences in the input document, while leaving their content unchanged. In such a case, *Leading sentences* would yield very different summaries depending on the reordering, and the ROUGE performance would approach that of *Random sentences*. A similar drawback affects the *Concept-frequency based summarizer* proposed in Ramanathan et al. (2009), which also leverages Wikipedia (see Section 4.2 for a quick overview), but relies on sentence ordering while extracting the summary. Despite high

Table 1
Average ROUGE-1 and ROUGE-2 scores.

Summarizer	Description	ROUGE-1	ROUGE-2
Leading sentences	First few sentences chosen as summary	0.45	0.22
Random sentences	Sentences chosen uniformly at random	0.41	0.15
Best sentences	Sentences chosen manually to match reference summaries	0.51	0.29
Wikipedia-based summarizer	Summarizer proposed in Section 4	0.46	0.23
Concept-frequency based summarizer	Summarizer proposed in Ramanathan et al. (2009)	0.47	0.23
MS Word	AutoSummarize feature in MS Word	0.46	0.18
S28	Top performing DUC 2002 summarizer	0.48	0.23
S21	Top performing DUC 2002 summarizer	0.48	0.22
S31	Top performing DUC 2002 summarizer	0.47	0.20
S29	Top performing DUC 2002 summarizer	0.46	0.21
S27	Top performing DUC 2002 summarizer	0.46	0.21

Table 2
Comparison of summarizers with similar ROUGE-1 scores.

Summarizer	ROUGE-1	Precision	Recall	F-measure	ROUGE-2
Leading sentences	0.45	0.53	0.45	0.46	0.22
Wikipedia-based summarizer	0.46	0.57	0.50	0.51	0.23
MS Word	0.46	0.33	0.35	0.32	0.18

ROUGE scores of 0.47 and 0.23 on the DUC 2002 data set, it is not robust to sentence reordering. This drawback is overcome in the proposed *Wikipedia-based summarizer* by ranking the sentences and concepts simultaneously, and using the sentence ranks to extract the summary.

Overall, we find that the *Wikipedia-based summarizer* performs competitively against the top systems from DUC 2002. In particular, it shows the highest ROUGE-2 score, and also significantly improves over the baseline techniques (*Random sentences* and *Leading sentences*).

Finally, it is interesting to note that the Wikipedia-based summarizer and the MS Word summarizer both show a ROUGE-1 score of 0.46. This prompted us to conduct a further evaluation using the precision and recall metrics. For this purpose, we use *Best sentences* as the benchmark, and compute the precision, recall and F-measure of a candidate summary as follows:

$$\text{Precision} = \frac{\text{Count}_{\text{match}}(\text{sentence})}{\text{Count}_{\text{candidate}}(\text{sentence})} \tag{14}$$

$$\text{Recall} = \frac{\text{Count}_{\text{match}}(\text{sentence})}{\text{Count}_{\textit{Bestsentences}}(\text{sentence})} \tag{15}$$

$$\text{F-measure} = \frac{2(\text{Precision})(\text{Recall})}{\text{Precision} + \text{Recall}} \tag{16}$$

where $\text{Count}_{\text{match}}(\text{sentence})$ represents the number of sentences present in both the candidate summary and *Best sentences* summary, $\text{Count}_{\text{candidate}}(\text{sentence})$ represents the total number of sentences present in the candidate summary, and $\text{Count}_{\textit{Bestsentences}}(\text{sentence})$ represents the total number of sentences present in the *Best sentences* summary. The individual scores are then averaged over the entire DUC 2002 dataset.

The results of our evaluation are shown in Table 2 for three candidate summarizers: *Leading sentences*, *Wikipedia-based summarizer*, and *MS Word summarizer*. These three candidates were chosen due to their near identical ROUGE-1 scores of 0.45, 0.46 and 0.46, respectively. Although the ROUGE-1 scores are very close, the summarizers can be distinguished based on the precision, recall, and F-measure. It is seen that the Wikipedia-based summarizer has the highest precision, recall and F-measure, while the MS Word summarizer performs the poorest. A similar behavior is seen in the corresponding ROUGE-2 scores.

To further confirm the above findings, we decided to manually assess the summary quality between the Wikipedia-based summarizer and the MS Word summarizer. We found several cases where the summaries were quite different, even though the ROUGE-1 scores are identical. As an example, we give below the summary of a news article on exoplanet discovery New Planet, first using the MS Word summarizer, and then using the Wikipedia-based summarizer. The sentences from the Wikipedia-based summarizer that are not picked up by the MS Word summarizer are shown in bold.

MS Word summarizer:

Red dwarfs are low-energy, tiny stars that give off dim red light and last longer than stars like our sun. Until a few years ago, astronomers did not consider these stars as possible hosts of planets that might sustain life. The discovery of the new planet, named 581 c, is sure to fuel studies of planets circling similar dim stars. About 80% of the stars near Earth are red dwarfs. The new planet is about five times heavier than Earth. If rocky like Earth, which is what the prevailing theory proposes, it has a diameter about 1 1/2 times bigger than our planet. If it is too thick, the planet's surface temperature could be too hot. The new planet seems just right. It does not mean there is life, but it means it is an Earth-like planet in terms of potential habitability. The new planet is probably full of liquid water, hypothesizes Stephane Udry, the lead author.

Wikipedia-based summarizer:

For the first time astronomers have discovered a planet outside our solar system that is potentially habitable, with Earth-like temperatures, a find researchers described Tuesday as a big step in the search for life in the universe. The planet is just the right size, might have water in liquid form, and in galactic terms is relatively nearby at 120 trillion miles away. And it is worth noting that scientists' requirements for habitability count Mars in that category: a size similar to Earth's with temperatures that would permit liquid water. Until a few years ago, astronomers did not consider these stars as possible hosts of planets that might sustain life. It does not mean there is life, but it means it is an Earth-like planet in terms of potential habitability.Because of its temperature and relative proximity, this planet will most probably be a very important target of the future space missions dedicated to the search for extraterrestrial life.

The difference in the above summaries is obvious, though the ROUGE-1 scores are identical, and in fact, only 10% below the upper bound of 0.51. This suggests that perhaps the ROUGE-1 score alone is not a sufficient indicator of summarizer performance.[13] It is

[13] Note that the total range of variation of the average ROUGE-1 scores shown in Table 1 is only 0.10. This is one reason why we have chosen to represent the ROUGE-1 scores only to a two decimal precision.

more appropriate to view the ROUGE-1 metric in conjunction with other metrics listed in Table 2 like precision, recall and ROUGE-2.

6.4. Incremental summarization

The incremental summarization property (see Definition 1 in Section 4.2) requires that longer summaries are generated in real-time by adding sentences to shorter summaries. We have built an application for browsing news articles, where the user is first presented with a short summary, after reading which, he/she can then interactively seek the required level of detail. In our implementation, the user is allowed to increment or decrement the summary size using hand gestures, or simply move onto the next article. Thus, the user could go in a single step or multiple steps from the initial summary to the full article. As discussed in Section 4.2, the incremental summarization property helps maintain the continuity of information flow over successive increments/decrements.

With the aim of determining the usefulness of incremental summarization, we conducted a study within a group of 30 users. Our test set consisted of 20 longish news articles, each over 1000 words. For each article, we first presented a 50 word summary, and then handed control to the user. The user was then allowed to iteratively increment or decrement the summary size any number of times, and view the resulting summary. We then asked each user to read the full article, and answer the following four questions:

1. Would you rather use incremental summarization or view a single summary? (yes/no)
2. Was the response to the increment/decrement fast enough? (yes/no)
3. How many articles (out of 20) are better read with incremental summarization?
4. Rate the quality of the presented summaries on a scale of 1–10, with 1 being least satisfactory and 10 being most satisfactory.

Almost the entire user group answered questions 1 and 2 in the affirmative. In response to question 3, we found that about 70% of the time, users felt that incremental summarization helped them better understand the news article. The average rating of summary quality was 7 out of 10.

6.5. Canonical examples

We conclude this section by providing 100-word summaries for two news articles, along with the top concepts returned by our Wikipedia-based summarizer.

Summary	Top concepts
Astronaut Luca Parmitano rushed to space station after spacesuit leaks (http://www.newsday.com/news/nation/astronaut-luca-parmitano-rushed-to-space-station-after-spacesuit-leaks-1.5706024)	
In one of the most harrowing spacewalks in decades, an astronaut had to rush back into the International Space Station on Tuesday after a mysterious water leak inside his helmet robbed him of the ability to speak or hear and could have caused him to choke or even drown. His spacewalking partner, American Christopher Cassidy, had to help him inside after NASA quickly aborted the spacewalk. In a late afternoon news conference, NASA acknowledged the perilous situation that Parmitano had found himself in, and space station operations manager Kenneth Todd promised to turn over every rock to make sure it never happens again.	Extra-vehicular activity Astronaut Apollo program Lyndon B. Johnson Space Center International Space Station
Bill Gates: the college lecture is dead but Microsoft Bob isn't (http://www.pcworld.com/article/2044399/bill-gates-the-college-lecture-is-dead.html)	
The implication is that students will use the Internet to begin learning online from major brands like MIT or the University of California, transforming the lecture from a live performance into something like the music industry, where most music is consumed in pre-recorded form. Gates has long been a proponent of using technology to solve the worlds problems, and of solutions like the online education services such as those offered by Salman Khan and the Khan Academy. Gates noted, however, that the problems facing impoverished nations throughout the globe often extend into several different areas; for example, the average IQ in sub-Saharan Africa is 82, he said.	Bill Gates Microsoft E-learning High school Khan Academy

7. Conclusion

The advent of human-generated knowledge bases like Wikipedia can help us understand the salient topics in a given text document. This can, in turn, help us extract summary sentences which convey these salient topics. In this paper, we studied one such approach that leverages Wikipedia in conjunction with graph-based summarization. We first constructed a bipartite sentence–concept graph, and proposed an iterative ranking algorithm for selecting summary sentences. Convergence of the iterative updates was studied under various scenarios: starting with undirected sentence–concept edges, and then allowing weighted and directed edges. Then, we looked at personalization and query-focusing of summaries, where the sentence ranking was dependent on user interests and queries, respectively. Finally, we extended our Wikipedia-based summarization to include multiple correlated text inputs.

A unique feature of our work is the introduction of the *incremental summarization* property, whereby both our single document and multi-document summarizers can provide additional content in real-time. Thus, users can first view an initial summary, and if interested, can then request additional content. We provided formal definitions of sentence-level and word-level incremental summarization, and used them as guiding factors in the design of our summarization algorithms. Finally, during performance evaluations, we noted how summarizers with identical ROUGE-1 scores could output vastly different summaries, thus motivating the need for further research into automatic evaluation methods. Indeed, a few other metrics (for example, the one based on Jensen–Shannon divergence Lin, Cao, Gao, & Nie (2006)) have been proposed over the years, but they rely on extensive training and the availability of reference summaries. Thus, it is fair to say that automated summary evaluation presents an interesting avenue for further research.

On the theoretical side, it would be interesting to conduct an axiomatic study of summarization algorithms that can leverage the sentence–concept mapping and satisfy the incremental summarization property. This is a topic for future work, both for single and multi-document summarization.

Appendix A. Proofs for Section 4.2

Proof of Theorem 1. Sentence Ranker 1 initializes the sentence scores as $x_1^{(0)} = x_2^{(0)} = \ldots = x_n^{(0)} = 1/\sqrt{n}$, and then repeats the following two steps:

Step 1: If $\mathbf{x}^{(k)}$ is defined for $k \geqslant 0$, then

$$y_j^{(k+1)} = \sum_{i:g_{ij}=1} x_i^{(k)} = \sum_i g_{ij} x_i^{(k)}, \quad \text{for } j \in \mathcal{M}. \tag{17}$$

The concept-score vector is now given by $\mathbf{y}^{(k+1)} = \left(y_1^{(k+1)} y_2^{(k+1)} \ldots y_m^{(k+1)}\right)^T$.
Step 2: Once $\mathbf{y}^{(k+1)}$ is defined, $\mathbf{x}^{(k+1)}$ is computed as follows:

$$\tilde{x}_i^{(k+1)} = \sum_{j:g_{ij}=1} y_j^{(k+1)} = \sum_j g_{ij} y_j^{(k+1)}, \quad \text{for } i \in \mathcal{N}, \tag{18}$$

and $\mathbf{x}^{(k+1)} = \tilde{\mathbf{x}}^{(k+1)}/\|\tilde{\mathbf{x}}^{(k+1)}\|$, i.e., $\mathbf{x}^{(k+1)}$ is the L^2-normalized version of $\tilde{\mathbf{x}}^{(k+1)}$.

It is easy to see from (17) and (18) that the iterations evolve as:

$$\begin{aligned} \mathbf{y}^{(k+1)} &= G^T \mathbf{x}^{(k)}, \\ \tilde{\mathbf{x}}^{(k+1)} &= G\mathbf{y}^{(k+1)}, \\ \mathbf{x}^{(k+1)} &= \frac{\tilde{\mathbf{x}}^{(k+1)}}{\|\tilde{\mathbf{x}}^{(k+1)}\|} \end{aligned} \tag{19}$$

Thus $\mathbf{y}^{(k+1)} = (G^T G)^k G^T \mathbf{x}^{(0)}$, and $\mathbf{x}^{(k+1)}$ is the unit vector in the direction of $(GG^T)^k \mathbf{x}^{(0)}$. Our aim now is to show that $\lim_{k\to\infty} \mathbf{x}^{(k)}$ exists, and to find this limit. To this end, we make an assumption that the largest eigenvalue of GG^T is unique, i.e., $|\lambda_1| > |\lambda_2| \geqslant |\lambda_3| \ldots \geqslant |\lambda_n|$, where λ_i denote the eigenvalues of GG^T with corresponding eigenvectors $\mathbf{q}_i$. Under this assumption, $\mathbf{q}_1$ is referred to as the principal eigenvector of GG^T. We now invoke the following result from linear algebra:

Lemma A.1. *If M is an $n \times n$ symmetric matrix and $\mathbf{q}_1$ is its principal eigenvector, then the L^2 unit vector in the direction of $M^k\mathbf{v}$ converges to $\mathbf{q}_1$ as $k \to \infty$, if $\mathbf{v}$ is not orthogonal to $\mathbf{q}_1$.*

Proof. Since M is symmetric, it admits a spectral decomposition

$$M = \sum_{i=1}^{n} \lambda_i \mathbf{q}_i \mathbf{q}_i^T.$$

Also, for any k, M^k has eigenvalues λ_i^k and corresponding orthonormal eigenvectors $\mathbf{q}_i$. Since the eigenvectors span the whole space,

$$\mathbf{v} = \sum_{i=1}^{n} c_i \mathbf{q}_i,$$

and thus,

$$M^k \mathbf{v} = \sum_{i=1}^{n} c_i \lambda_i^k \mathbf{q}_i.$$

The L^2 unit vector in the direction of $M^k\mathbf{v}$ is

$$\mathbf{u} = \sum_{i=1}^{n} \frac{c_i \lambda_i^k \mathbf{q}_i}{\left(\sum_{j=1}^{n} c_j^2 \lambda_j^{2k}\right)^{1/2}}.$$

The proof is completed by using the fact that $c_1 \neq 0$, dividing the numerator and denominator of the above fraction by $c_1\lambda_1^k$ and letting $k \to \infty$.

We invoke Lemma A.1 with $M = GG^T$, and $\mathbf{v} = \mathbf{x}^{(0)}$. But first, we need to verify that $\mathbf{x}^{(0)}$ is not orthogonal to $\mathbf{q}_1$, the principal eigenvector of GG^T. This follows from the fact that each entry of $\mathbf{x}^{(0)}$ and $\mathbf{q}_1$ is non-negative. The latter is a direct consequence of Lemma A.1, in conjunction with the fact that each entry of G (and hence M) is non-negative.

Thus, we find that $\mathbf{x}^{(k+1)}$ converges to the principal eigenvector of GG^T. Since $\mathbf{y}^{(k+1)} = G\mathbf{x}^{(k)}$, it also follows that $\lim_{k\to\infty}\mathbf{y}^{(k)}$ exists, and is a scaled version of the principal eigenvector of G^TG. We remark here that additional normalization of the concept scores does not affect our analysis, since normalization is only a scalar multiplication. The only difference would be that $\mathbf{y}^{(k)}$ would then actually converge to the principal eigenvector of G^TG.

Proof of Proposition 3. As before, let $\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_n$ denote the n eigenvectors of GG^T. Then, Theorem 1 states that $\lim_{k\to\infty}\mathbf{x}^{(k)} = \mathbf{q}_1$. The proof is now completed upon showing that $\mathbf{x}^\star = \mathbf{q}_1$. This is done as follows:

$$\max_{\|\mathbf{x}\|=1} \left(\|G^T\mathbf{x}\|\right)^2 = \max_{\|\mathbf{x}\|=1} \mathbf{x}^T GG^T \mathbf{x} \tag{20}$$

$$= \mathbf{x}^T \left(\sum_{i=1}^{n} \lambda_i \mathbf{q}_i \mathbf{q}_i^T\right)\mathbf{x} \tag{21}$$

$$= \frac{\sum_{i=1}^{n} \lambda_i \left(\mathbf{x}^T\mathbf{q}_i\right)^2}{\mathbf{x}^T\mathbf{x}} \tag{22}$$

$$= \frac{\sum_{i=1}^{n} \lambda_i \left(\mathbf{x}^T\mathbf{q}_i\right)^2}{\mathbf{x}^T\left(\sum_{i=1}^{n} \mathbf{q}_i\mathbf{q}_i^T\right)\mathbf{x}} \tag{23}$$

$$= \frac{\sum_{i=1}^{n} \lambda_i \left(\mathbf{x}^T\mathbf{q}_i\right)^2}{\sum_{i=1}^{n} \left(x^T\mathbf{q}_i\right)^2} \tag{24}$$

$$\leqslant \lambda_1, \tag{25}$$

where (20) follows from the definition of L^2 norm, (21) uses the spectral decomposition of GG^T, and (22), (23), (24) leverage the orthonormal property of eigenvectors $\mathbf{q}_i$. Equality in (25) is attained when $\mathbf{x} = \mathbf{q}_1$.

Appendix B. Connection with latent topic models

Sentence Ranker 1 is based on mutual reinforcement of sentences and concepts through the iterative updates (4). In Appendix A, the vector of sentence and concept scores were shown to converge to the principal eigenvectors of GG^T and G^TG, respectively. One important question is why should we take the weights assigned by the principal eigenvector of GG^T (resp. G^TG) as the representative weight for the sentences (resp. concepts). In what follows, we take a step towards answering this question, and show that under a bag-of-words model, the generative weights assigned to the concepts is in the same relative order as the dominant eigenvector of G^TG. Before getting into the details, we note the following simple lemma which we will invoke later:

Lemma B.1. *Define*

$$f(x,y) = 1 - (1-x)^a - (1-y)^a + (1-x-y)^a.$$

If $a > 1$, then $f(x,y)$ is an increasing function in x for fixed y.

Proof. Follows by noting that:

$$\frac{\partial}{\partial x} f(x,y) = a((1-x)^{a-1} - (1-x-y)^{a-1}) \geq 0.$$

Suppose each sentence is made up of b words, and each word is generated according to some topic model. Suppose, for simplicity, that we know the topics themselves. So, for a sentence s_i, we know the constituent topics $t_{s,1}, t_{s,2}, \ldots, t_{s,b}$. Let $z_1, z_2, \ldots, z_m$ be the m topics which appear with probabilities $\theta_1, \theta_2, \ldots, \theta_m$ in the document. Assume without loss of generality that the θ_i's are ordered. For notational ease, we denote

$$\{z_i \in s\} \Longleftrightarrow \{t_{s,j} = z_i \quad \text{for some } j\}.$$

Next consider the bipartite graph that maps sentences to the Wikipedia concepts. For this bipartite graph, given a subset of sentences $\mathcal{U} = \{s_{i_1}, s_{i_2}, \ldots, s_{i_d}\}$, and concept weights $p_{z_1}, p_{z_2}, \ldots, p_{z_m}$, the cut norm is given by

$$\|\mathcal{U}\|_{cut} = \sum_{l=1}^{d} \sum_{j: z_j \in s_{i_l}} p_{z_j}.$$

While the iterative updates work on sentence–concept reinforcement, one can talk about another summarization approach where to each topic we assign the weights θ_i (instead of the weights from the iterative ranking algorithm), and then maximize the cut norm of the resulting bipartite graph. The latter approach is actually similar to the iterative updates, in the sense that if to sentence s we assign weight $w_s = \sum_{z_i \in s} \theta_i$ and pick the highest ranked sentences, we will in turn maximize the cut norm.

In what follows, we show that when we have sufficiently large number of sentences, with a high probability, the observed principal eigenvector of $G^T G$ will have the same relative ordering as $\boldsymbol{\theta}$. Thus, it does not matter if we are maximizing the cut norm or using the iterative updates: the summary generated will be the same with a high probability.

Lemma B.2. *As the number of sentences $n \to \infty$, the probability that the coordinates of the dominant eigenvector of $G^T G$ will have the same relative ordering as $\boldsymbol{\theta}$ tends to one.*

Proof. For any sentence s,

$$p_i := P(z_i \in s) = 1 - (1-\theta_i)^b.$$

Also, for two distinct topics z_{i_1}, z_{i_2}, we have

$$p_{i_1 i_2} := P(z_{i_1}, z_{i_2} \in s) = 1 - (1-\theta_{i_1})^b - (1-\theta_{i_2})^b + (1-\theta_{i_1}-\theta_{i_2})^b.$$

Using Lemma B.1, we note that for $i < j, p_{il} > p_{jl}$.

Lets take a look at the $G^T G$ matrix. Denote by $c_{ij}^{(k)}$ the ijth entry of $(G^T G)^k$. Then, for any $1 \leqslant i \leqslant m$

$$c_{ii}^{(1)} = \mid \{j : 1 \leqslant j \leqslant n, S_j \leftrightarrow z_i\} \mid .$$

Clearly $c_{ii}^{(1)} \sim \text{Bin}(n, p_i)$ for all $1 \leqslant i \leqslant m$. Next, note that for $i \neq j$

$$c_{ij}^{(1)} = \mid \{l : 1 \leqslant l \leqslant n, z_i, z_j \in S_l\} \mid$$

Again,

$$c_{ij}^{(1)} \sim \text{Bin}(n, p_{ij}).$$

If, $E = E(G^T G)/n$ then $E_{ii} = p_i$ and $E_{ij} = p_{ij}$ for $i \neq j$. For any $\epsilon > 0$, by the weak law,

$$P\left(\left\|\frac{1}{n} G^T G - E\right\| > \epsilon\right) \to 0 \text{ as } n \to \infty,$$

where $||\cdot||$ stands for the Frobenius or the matrix L_2 norm. Suppose $x^{(0)} = (\frac{1}{n}, \frac{1}{n}, \ldots, \frac{1}{n})$. Then the ith coordinate of $y^{(0)} = G^T x_0$ is

$$y_i^{(0)} = \frac{\mid \{s : z_i \in s\} \mid}{n}.$$

For n large enough, with a large probability,

$$y^{(0)} \approx \mathbf{p}.$$

If we use E as a proxy for $G^T G/n$, then our goal is to show that $E^k \mathbf{p}$ has the coordinates in the same relative order as in $\boldsymbol{\theta}$ or equivalently, $\mathbf{p}$.

Assume that we have a vector $\mathbf{z}$ so that $z_i \geqslant z_j \geqslant 0$ for all $i \leqslant j$. We will show that if $\tilde{\mathbf{z}} = E\mathbf{z}$ then $\tilde{z}_i \geqslant \tilde{z}_j$ for all $i \leqslant j$. To prove this claim, we first take $i = 1,\ j = 2$. Note that $p_{1j} \geqslant p_{2j}$ for all $j > 2$ because of Lemma B.1. Thus we readily observe that

$$p_{1j} z_j \geqslant p_{2j} z_j \quad \text{for all } j > 2. \tag{26}$$

Next, using $p_1 \geqslant p_2$ and $z_1 \geqslant z_2$, observe that

$$p_1 z_1 + p_{12} z_2 - (p_{12} z_1 + p_2 z_2) = p_1 z_1 - p_2 z_2 - p_{12}(z_1 - z_2) \geqslant p_2(z_1 - z_2) - p_{12}(z_1 - z_2) = (z_1 - z_2)(p_2 - p_{12}) \geqslant 0. \tag{27}$$

Using (26) and (27), we have that

$$\tilde{z}_1 = \sum_{j=1}^{m} E_{1j} z_j \geqslant \sum_{j=1}^{m} E_{2j} z_j = \tilde{z}_2.$$

Of course, there is nothing special about comparing the first and second indices. Using a similar argument, we can easily see that as long as $i \leqslant j$ and hence $p_i \geqslant p_j$ and $z_i > z_j$, we will always have

$$\tilde{z}_i \geqslant \tilde{z}_j.$$

Since $y_i^{(0)} \geqslant y_j^{(0)}$ for all $i \leqslant j$, we have, by induction, that

$$y_i^{(k)} \geqslant y_j^{(K)} \quad \text{when } i \leqslant j.$$

Hence, the ordering of weights is preserved from the θ vector to the $y^{(k)}$ vector with a high probability if n is sufficiently large. Thus in the cut norm, if we give the existing edges of the sentence-topic bipartite graph weight one, then picking sentences with higher $\mathbf{x}$ weights is equivalent to maximizing the cut norm.

Remark. Suppose we assume the bag-of-words model, and each sentence is made up of b concepts where b is large. In that case, with probability nearing one, each sentence will contain all the concepts, and thus, they will be identical as far as the sentence–concept bipartite graph is concerned. Essentially, the sentences being *i.i.d* makes it hard to distinguish them. This is one of the weaknesses of the bag-of-words assumption.

References

Agrawal, R., Gollapudi, S., Halverson, A., & Ieong, S. (2009). Diversifying search results. In *Proceedings of the second ACM international conference on web search and data mining* (pp. 5–14). ACM.

Angheluta, R., De Busser, R., & Moens, M. -F. (2002). The use of topic segmentation for automatic summarization. In *Proc. DUC 2002*.

Barzilay, R., Elhadad, N., & McKeown, K. R. (2001, March). Sentence ordering in multidocument summarization. In *Proceedings of the first international conference on human language technology research* (pp. 1–7). Association for Computational Linguistics.

Bawakid, A., & Oussalah, M. (2010). Summarizing with Wikipedia. In *Proceedings of the text analysis conference (TAC)*.

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual web search engine. *Computer Networks and ISDN Systems, 30*(1), 107–117.

Brunn, M., Chali, Y., & Dufour, B. (2002). University of Lethbridge. The University of Lethbridge text summarizer at DUC 2002. In *Proc. DUC 2002*.

Clarke, C. L., Kolla, M., Cormack, G. V., Vechtomova, O., Ashkan, A., Bttcher, S., et al. (2008, July). Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval* (pp. 659–666).

Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2009). *Introduction to algorithms* (3rd ed.). MIT press.

Das, D., & Martins, A. F. (2007). A survey on automatic text summarization. *Literature Survey for the Language and Statistics II Course at CMU, 4*, 192–195.

Document Understanding Conference 2002. <http://www-nlpir.nist.gov/projects/duc/guidelines/2002.html>.

Document Understanding Conferences 2001–2007. <http://duc.nist.gov/>.

El-Arini, K., Veda, G., Shahaf, D., & Guestrin, C. (2009). Turning down the noise in the blogosphere. In *Proceedings of the 15th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 289–298). ACM.

Erkan, G., & Radev, D. R. (2004). LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research (JAIR), 22*, 457–479.

Filatova, E., & Hatzivassiloglou, V. (2004, August). A formal model for information selection in multi-sentence text extraction. In *Proceedings of the 20th international conference on computational linguistics* (p. 397). Association for Computational Linguistics.

Gabrilovich, E., & Markovitch, S. (2006, July). Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the national conference on artificial intelligence* (Vol. 21(2), p. 1301). Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Gong, S., Qu, Y., & Tian, S. (2010). Summarization using Wikipedia. In *Proceedings of text analysis conference* (Vol. 2010).

Hirao, T., Sasaki, Y., Isozaki, H., & Maeda, E. (2002). NTT's text summarization system for DUC-2002. In *Proc. DUC 2002*.

Khuller, S., Mossb, A., & Naor, J. (1999). The budgeted maximum coverage problem. *Information Processing Letters, 70*, 39–45.

Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *Journal of the ACM (JACM), 46*(5), 604–632.

Krause, A., & Guestrin, C. (2005). A note on the budgeted maximization of submodular functions. *Tech. Report CMU-CALD-05-103*.

Lin, C. Y. (2004). ROUGE: A package for automatic evaluation of summaries. *Workshop on text summarization branches out (WAS 2004)*, Barcelona, Spain, July 25–26, 2004.

Lin, H., Bilmes, J., & Xie, S. (2009). Graph-based submodular selection for extractive summarization. In *IEEE Workshop on Automatic speech recognition & understanding 2009. ASRU 2009* (pp. 381–386). IEEE.

Lin, C.Y., & Hovy, E. (2003, May). Automatic evaluation of summaries using n-gram co-occurrence statistics. In *Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology* (Vol. 1, pp. 71–78). Association for Computational Linguistics.

Lin, C.Y., Cao, G., Gao, J., & Nie, J. Y. (2006, June). An information-theoretic approach to automatic evaluation of summaries. In *Proceedings of the main conference on human language technology conference of the North American chapter of the association of computational linguistics* (pp. 463-470). Association for Computational Linguistics.

Lin, H., & Bilmes, J. (2010, June). Multi-document summarization via budgeted maximization of submodular functions. In *North American chapter of the association for computational linguistics/human language technology conference (NAACL/HLT-2010)*, Los Angeles, CA.

Mani, I., & Maybury, M. T. (Eds.). (1999). *Advances in automatic text summarization*. MIT press.

McDonald, R. (2007). A study of global inference algorithms in multi-document summarization. *Advances in Information Retrieval*, 557–564.

McKeown, K. R., Klavans, J. L., Hatzivassiloglou, V., Barzilay, R., & Eskin, E. (1999, July). Towards multidocument summarization by reformulation: Progress and prospects. In *Proceedings of the national conference on Artificial intelligence* (pp. 453–460).

Miao, Y., & Li, C. (2010). WikiSummarizer – A Wikipedia-based summarization system (2010). In *Proc. text analysis conference (TAC)*.

Mihalcea, R., & Tarau, P. (2005). A language independent algorithm for single and multiple document summarization. In *Proceedings of IJCNLP* (Vol. 5).

Mihalcea, R., & Tarau, P. (2004). TextRank: Bringing order into texts. In *Proceedings of EMNLP* (Vol. 4, pp. 404–411).

Potentially habitable planet found outside solar system. <http://www.ufoevidence.org/news/article357.htm>.

Pourvali, M., & Abadeh, M. S. (2012a). A new graph based text segmentation using Wikipedia for automatic text summarization. *International Journal of Advanced Computer Science and Applications (IJACSA), 3*(1).

Pourvali, M., & Abadeh, M. S. (2012b). Automated text summarization base on lexicales chain and graph using of WordNet and Wikipedia knowledge base. *International Journal of Computer Science Issues (IJCSI), 9*(1). No. 3.

Radev, D. R., Hovy, E., & McKeown, K. (2002). Introduction to the special issue on summarization. *Computational Linguistics, 28*(4), 399–408.

Radev, D. R., Jing, H., Stys', M., & Tam, D. (2004). Centroid-based summarization of multiple documents. *Information Processing & Management, 40*(6), 919–938.

Ramanathan, K., Sankarasubramaniam, Y., Mathur, N., & Gupta, A. (2009). Document summarization using Wikipedia. In *Proceedings of the first international conference on intelligent human computer interaction* (pp. 254–260). India: Springer.

ROUGE package for evaluating summaries. <http://www.berouge.com/Pages/default.aspx>.

Schlesinger, J. D., Conroy, J. M., Okurowski, M. E., Wilson, H. T., O'Leary, D. P., Taylor, A., et al. (2002). Understanding machine performance in the context of human performance for multi-document summarization. In *Proc. DUC 2002*.

Seneta, E. (1981). *Non-negative matrices and Markov chains*. New York: Springer.

Skoutas, D., Minack, E., & Nejdl, W. (2010). Increasing diversity in web search results. In *Proc. WebSci10: Extending the frontiers of society on-line*, April 26–27, 2010, Raleigh, NC, US.

Strang, G. (1988). *Linear algebra and its applications* (3rd ed.). New York: Academic. 19802.

Takamura, H., & Okumura, M. (2009, March). Text summarization model based on maximum coverage problem and its variant. In *Proceedings of the 12th conference of the European chapter of the association for computational linguistics* (pp. 781–789). Association for Computational Linguistics.

van Halteren, H. (2002). University of Nijmegen. Writing style recognition and sentence extraction. In *Proc. DUC 2002*.

Wan, X., Yang, J., & Xiao, J. (2006). Incorporating cross-document relationships between sentences for single document summarizations. *Research and Advanced Technology for Digital Libraries*, 403–414.

Ye, S., Chua, T. S., & Lu, J. (2009, August). Summarizing definition from Wikipedia. In *Proceedings of the joint conference of the 47th annual meeting of the ACL and the 4th international joint conference on natural language processing of the AFNLP* (Vol. 1, pp. 199–207). Association for Computational Linguistics.

Yeh, J. Y., Ke, H. R., & Yang, W. P. (2008). iSpreadRank: Ranking sentences for extraction-based summarization using feature weight propagation in the sentence similarity network. *Expert Systems with Applications, 35*(3), 1451–1462.

Zhai, C. X., Cohen, W. W., & Lafferty, J. (2003). Beyond independent relevance: Methods and evaluation metrics for subtopic retrieval. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval* (pp. 10–17). ACM.