Corpus-based Content Construction

Balaji Vasan Srinivasan, Pranav Ravindra Maneriker, Kundan Krishna, Natwar Modani
Adobe Research Big data Experience Lab
Bangalore, India
[balsrini,pmanerik,kkrishna,nmodani]@adobe.com

Abstract

Enterprise content writers are engaged in writing textual content for various purposes. Often, the text being written may already be present in the enterprise corpus in the form of past articles and can be re-purposed for the current needs. In the absence of suitable tools, authors manually curate/create such content (sometimes from scratch) which reduces their productivity. To address this, we propose an automatic approach to generate an initial version of the author's intended text based on an input content snippet. Starting with a set of extracted textual fragments related to the snippet based on the query words in it, the proposed approach builds the desired text from these fragment by simultaneously optimizing the information coverage, relevance, diversity and coherence in the generated content. Evaluations on standard datasets shows improved performance against existing baselines on several metrics.

1 Introduction

Due to the increase in the number of channels and the need for more customized content for different audience segments, content authors need to produce content at a very rapid pace. On the other hand, a lot of such content has already been produced and is available in enterprise content repositories. Ideally, the author looks for such content and re-purposes them for her current need. Due to the tedium of this task, the author may resort to creating the content from scratch. Automating this task can greatly aid the author in improving their overall productivity.

Given an enterprise author's requirement in the form of a snippet describing the content she is trying to build, standard querying mechanisms can be utilized to fetch the right set of articles that can cater to her needs. However, constructing an initial piece of 'article' that can cater to her needs would still be a non-trivial task. Existing work in this direction focuses on generating content for standard platforms like Wikipedia (Banerjee and Mitra, 2016), where the target content is divided into different 'categories/section' and text is generated for each section. While this could be useful, the absence of a predefined structure in an enterprise setting makes the extension of such an approach challenging. This calls for an approach that can automatically build an author's content **covering various aspects** of the target article.

The content style in an enterprise corpus is fairly consistent. However, there could be multiple representations of the same information across the repository. This necessitates the content generation to **reduce the redundancy** in the constructed content and adding topical diversity to the article. Finally, the content generated should be coherent with a **cohesive flow** of information.

To address all these requirements, we propose an approach that automatically generates a *coherent* and *non-redundant* article based on an author's input snippet simultaneously optimizing the *relevance* of the generated text to the snippet, *coverage* of different aspects of the content and *diversity* in the information content. The acceptance criteria for such systems can be a reasonable draft instead of publishable content, given that the objective of such a system is not to replace the authors, but to provide them with a draft that can be polished quickly. Table 1 shows a sample article generated by the proposed algorithm based on an input snippet from DUC2007 data (Dang, 2007).

Proceedings of the 27th International Conference on Computational Linguistics, pages 3505–3515
Santa Fe, New Mexico, USA, August 20-26, 2018.

Table 1: A sample snippet and the corresponding article generated by the proposed algorithm based on the DUC2007 data (Dang, 2007).

Input Snippet: Microsoft's antitrust problems and its alleged illegal behavior and antitrust proceedings against the company.
Generated article by CCC: a justice department spokesman confirmed thursday that we have an ongoing investigation and its continuing. microsoft based in redmond of washington has been the focus of antitrust investigations in the past years. microsoft corp is making a last minute offer to settle the charges. nbc news reported wednesday that the justice department and at least 19 states plan to file historic antitrust cases thursday against microsoft. if they proceed the suits could be among the biggest corporate challenges in history according to analysts. on monday in washington dc the federal government and 20 states kicked off the trial that accused microsoft of illegal activities including bullying competitors. back in seattle washington state where microsoft is based the company fired off five upbeat productoriented press releases and other announcements. its web site on monday carried two articles in favor of microsoft. it helps the atmospherics of the governments case but its not clear beyond that what the effect will be. meanwhile the antitrust trial is already well under way and judge thomas penfield jackson is expected to rule early next year. but sherlund noted the ruling does rein microsoft in and it clearly comes at an awkward time for the company given the antitrust case. antitrust experts said that most countries still investigating microsofts practices were taking a waitandsee stance toward the company pending the outcome of the major us antitrust case

Our algorithm takes author intent in the form of a short snippet (say a sentence) and finds relevant content from a repository to create a pool of content. It then identifies the most representative part of the content (factoring in relevance level) and generates multiple candidate sentences by compressing information from multiple related sentences to enrich the information content (and reduce redundancy). The compression algorithm leverages usage patterns in the repository to form sentences with good readability. Subsequent candidate sentence generations account for the already generated candidates to ensure diversity. We finally assemble the content in an appropriate order by solving an integer linear program that simultaneously optimizes the coherence with various linguistic quality measures to determine the best ordering. Thus, our system strives to generate a draft of content that is relevant to the author intent, diverse, information rich, cohesive and readable.

We present the results of performance evaluation of our system with DUC 2007 (Dang, 2007) and Australian Legal (Galgani et al., 2012) datasets. Our experiments show that our system produces drafts which are as good as state-of-art baselines for content generation in Rouge scores, diversity, readability and cohesiveness, and beat each of the baselines in at least one of these metrics.

2 Related Work

There is a significant body of work in the domain of **text summarization** (Nenkova and McKeown, 2011). The problem of corpus based content construction could be perceived as a summarization task once we have identified the required candidate content fragments from the corpus. In particular, the problem could be cast as a multi-document summarization or query-focused summarization. However, in our problem, there is no requirement to maintain the distribution of concepts/information between input document and output content, thus making it different from standard summarization problems.

There have been some efforts towards extending multi-document summarization for expanding a piece of content. Banerjee et al. (2015) use a sentence ranking and clustering formulation along with ILP based constraint optimization to create expanded documents. Bing et al. (2015) explore a more fine-grained setting, constructing sentences using noun/verb phrase level breakdown of input text. However, neither of these two approaches handle the document coherence problem at a granular level and rely on using the original sentence order as their notion of coherence.

Query-focused multi-document summarization approaches such as those of Li and Li (2014) and Wang et al. (2013) do not have an explicit model of coherence, in spite of using coherence as one of the evaluation criteria. In the proposed algorithm, we use an Integer Linear Program formulation to achieve a coherent generation despite not explicitly selecting contiguous content fragments.

Srinivasan et al. (2017) extend standard extractive summarization techniques to build article by combining different paragraphs in a corpus. However, they do not account for coherence or information redundancies in the generated content. Often, retaining an entire paragraph might not yield the best

information coverage. We explicitly account for these aspects in our proposed algorithm.

There are some recent efforts towards accounting for coherence in summarization by extending the transition probability based approach by Lapata and Barzilay (2005) and including coherence in their optimization frameworks. Parveen et al. (2016) use 'patterns' from a standard corpus to define the transition probability, while Nayeem and Chali (2017) and Wang et al. (2016) use information overlap based metric to define the transitions. Our framework also uses an approach similar to the latter to account for coherence in our text generation.

Knowledge enhancement: There have also been some efforts towards content elaboration in the context of knowledge enhancement. Schlaefer et al. (2011) aim to enhance question-answering by expanding a 'seed document' using web resources to construct the answer based on 'paragraph nuggets'. However, both lexical and semantic redundancies are undesirable for a content author. In our approach, we minimize the lexical and semantic redundancies based on a graph based sentence compression which jointly optimizes for relevance and diversity of the expanded content.

The work by Bairi et al. (2016) uses expansion of short documents to enhance the document representation for improved document classification. Their framework aims to improve performance for a task without using task-specific heuristics. However, since the objective is to maximize the classification performance, the algorithm proposed by Bairi et al. (2016) does not care about linguistic aspects of the elaborated content.

Taneva and Weikum (2013) identify relevant text snippets ('gems') by using an integer linear program to maximize relevance of selected words, and prefer the selection of contiguous content pieces since this maximizes the coherence of the content resulting in a lot of content being chosen from the same source. However, the information for a specific article can come from multiple sources and hence this will not achieve information coverage. We use this as one of the baselines for our comparisons and show improvements in the coverage of our proposed approach.

Wikipedia Generation: Mihalcea and Csomai (2007) identify key concepts in a document and link them to corresponding Wikipedia pages. While this helps to identify relevant Wikipedia areas for the document, it will not help in expanding the seed content from these sources. Sauper and Barzilay (2009) came up with the first complete version of such a system. Their work leverages the structure of Wikipedia articles of a category to produce a new article in the same category. Banerjee and Mitra (2016) extend these ideas to construct a Wikipedia article by learning the article structure from the structure of related articles, rather than relying on a single category. While this is closer to what we would like to achieve, the lack of structure in an enterprise article poses a unique challenge in our case. Further, if the corpus is huge, this can result in a lesser information coverage. We address this in our proposed algorithm via an iterative reward framework to maximize the information coverage of the generated article.

3 CCC: Corpus based Content Construction

We propose a **C**orpus-based **C**ontent **C**onstructor (CCC) that simultaneously optimizes for relevance to the input snippet, covers different information about the target content, avoids redundant sentences and improves the coherence of the final content. The algorithm involves 3 stages: extracting the intent of author from the snippet, identifying and **generating candidate content** towards the author's need and **assembling the generated content** in a coherent fashion.

We begin with extracting the top-k keywords in the snippet using the inverse document frequency (IDF) of the words in the corpus, thus capturing the most significant keywords in the snippet with respect to the corpus. A query q is then formulated by concatenating these 'k' keywords (Aula, 2003), where the choice of k determines the relevance and the amount of content that is available and fetched from the repository. A lower value of k results in the query under-representing the content and fetching articles that may not be very relevant. On the other hand, a higher value of k will result in a very specific query that might not fetch many results from the repository.

The retrieved fragments are broken into sentences. To select the appropriate content for construction, we create a graph and represent the sentences as nodes in the graph. The information content of the sentences is assigned as a "reward" to the corresponding nodes and their information overlap as the edge

weight. As explained later, the most "rewarding" part of the graph is identified and the sentences in that sub-graph are compressed to prepare candidates to be included in the final content. The sentence compression reduces the syntactic and lexical redundancies arising due to multiple manifestations of the same information in different parts of the corpus. Once the candidates are generated, the rewards of the nodes in the graph are adjusted to account for the information in the generated candidates. This allows for increased information coverage in the generated sentences by discounting the information covered by the previously extracted candidates.

The candidates are generated iteratively till the overall information relevant to the input snippet is exhausted in the graph. To select the final content, an Integer Linear Program (ILP) is formulated that maximizes the relevance to the snippet along with the information cohesion in the generated content. The optimization problem is also constrained to keep the constructed content to a certain budget (length) and avoid simultaneous selection of semantically very-similar sentences. The ILP outputs a set of compressed sentences along with their sequence to yield a cohesive content.

The candidate generation and coherent candidate organization are the key contributions of our algorithm and we describe the details below.

3.1 Candidate Generation via Sentence Compression:

The candidate set of fragments is broken into sentences and a graph $G(v, e)$ is constructed with every sentence as a node $v_i \in V$ in the graph. An initial "reward" r_i^0 for v_i is assigned based on its similarity to the query q. The edges $e \in E$ are constructed between every pair of fragment that contain a significant information overlap and the similarity between the sentences is assigned as its edge weight w_{ij}.

We iteratively select a sub-graph in G whose nodes can be used as candidates for compression. Each sub-graph consists of a node v_i along with its $1-$hop neighbors. The gain G_{v_i} of processing a node v_i (along with its $1-$hop neighbors) for sentence compression at round l is given by,

$$G_{v_i}^l = r_i^{l-1} + \sum_{v_j \in N_i} r_j^{l-1} \times w_{ij} \tag{1}$$

where N_i refers to the neighbors of node v_i. At step l, we select the v_i^* that has the maximum gain G_{v_i}. The sentence corresponding to the node selected v_i along with the sentences in its $1-$hop neighborhood is put into a set s_{v_i}, which is used for multi-sentence compression (Filippova, 2010) to remove the redundancy across the multiple sentences and generate candidates.

Multi-sentence compression constructs a (different) graph ($\hat{G}_c$) out of the candidates such that each word is a node, and each sentence represents a directed path in the graph. Specialized nodes serve as start-of-sentence and end-of-sentence nodes. A single node is used to represent all occurrences of a word with the same POS tag. The edges between nodes have weights representing the number of times the ordered combination of those node-words occurs across all sentences in s_{v_i}. The shortest paths (normalized by path length) are identified and the top-K generated sentences are used for further processing. For our purposes, we restrict the minimum number of words per generated sentence to 10 words and pick 1 sentence in every iteration.

A reward $r_{\hat{k}}$ is assigned to every compressed sentence based on its similarity to the query q. In order to factor the information captured by the compressed sentences for subsequent selections, the rewards of vertices v_i in the original graph ($G(v, e)$) whose information overlap ($w_{i\hat{k}}$, computed in the same way as the edge weights) with the compressed sentences is significant, is updated as as $r_i^{l+1} = r_i^l \times (1 - w_{i\hat{k}})$. This reduces the rewards for sentences which are covered by current set of compressed sentences thus reducing the chance of their inclusion in subsequent compression. This ensures that the information covered by the subsequent compression is different from already generated sentences while still relevant to the input query.

The candidate generation is stopped when there are no nodes left with significant reward (greater than a threshold) resulting in the same sub-graph being selected for successive compression.

3.2 Coherent Candidate Organization via ILP

The sentence compression yields candidates that cover the information space relevant to the input snippet. However, the compressed sentences might be grammatically incorrect and not ordered appropriately to be coherent. Therefore we select the right set of compressed sentences along with their order via an Interger Linear Program (ILP) formulation by extending the framework used by Deshpande et al. (2007) and Banerjee and Mitra (2016). The ILP objective is given by,

$$J = \sum_{i=1}^{K} w_i x_i + \lambda \sum_{i,j|coh_{ij}>\sigma} coh_{i,j} y_{i,j} \tag{2}$$

$$\text{such that: } \sum_{i=1}^{K} c_i x_i \leq B \tag{3}$$

$$y_{i,j} = 0 \text{ if } coh_{ij} < \sigma \text{ or } i = j \tag{4}$$

$$x_i + x_j \leq 1, \forall i, j | sim_{i,j} > 0.7 \tag{5}$$

$$\sum_i y_{s,i} = 1, \sum_i y_{i,e} = 1 \tag{6}$$

$$\sum_i y_{i,j} = \sum_i y_{j,i}, \sum_i y_{i,j} + \sum_i y_{j,i} = 2x_j \tag{7}$$

The binary variable x_i indicates the selection/non-selection of a compressed sentence i and the binary variable $y_{i,j}$ indicates a transition from a sentence x_i and x_j. Traversing the path of the sentences via $y_{i,j}$ would yield the final generated sentence. The w_i for each compressed sentence indicates a combination of the sentence's relevance to the snippet along with its overall linguistic quality. This ensures that the compressed sentence selected is not noisy and is relevant to what the author is looking to build. The second term in Eq. 2 maximizes the coherence of the selected flow of sentences. This follows from the work by Lapata and Barzilay (2005) which defines a model of coherence based on the topic overlap between successive sentences.

The constraint in Eq. 3 accounts for the length requirement of the author and limits the generated content to a target budget B. Eq. 4 prohibits a flow of content between lesser coherent sentences and avoids cycles in the arcs. Eq. 5 limits simultaneous selection two sentences with higher information overlap to produce less redundant content.

Similar to Banerjee and Mitra (2016), we constrain the optimization problem to also account for the overall quality of the generated content. The formulations to constrain the arcs is extended from WikiWrite (Banerjee and Mitra, 2016) by introducing dummy start s and end e nodes to the graph of sentences. An arc between two nodes exists if both the sentences are selected for the final content and are contiguous. The dummy nodes s and e are used to connect to the start and end sentences. Eq. 6 limits the number of starting and ending sentences to 1. Eq. 7 limit the number of incoming and outgoing arcs from a selected node to 1 each forcing a path via the arcs that indicate the flow in the selected content. Algorithm 1 summarizes the complete algorithm.

4 Evaluating different stages in CCC

In CCC, sentence compression (Filippova, 2010) is used to avoid redundant information in the constructed content by compressing information from multiple sentences into a single sentence. Further, the graph based candidate selection for compression enables covering diverse aspects around the target content. Finally, the ILP formulation facilitates coherence in the generated content.

To test if the different stages achieve the targeted goals, we test different parts of our algorithm on the DUC2007 Multi-document Summarization data (Dang, 2007). This data consists of a topic along with a set of documents related to the topic and reference summaries for the set of documents. For the sake of this experiment, we bypass the query construction stage. We use the topic associated with the document

Algorithm 1: Corpus-based Content Construction

Input: q: Snippet input by author, $\mathcal{F}$: Set of all fragments
Output: C: Generated Expanded Content
K $\leftarrow$ Top-k keywords from q based on IDF from $\mathcal{F}$;
F $\leftarrow$ query $\mathcal{F}$ with K;
$\mathcal{S} \leftarrow$ split F into sentences;
begin
  $\mathcal{G} \leftarrow$ graph($\mathcal{V}$, $\mathcal{E}$) where,
  $\mathcal{V} \leftarrow \mathcal{S}$, with reward $r_{v_i} \leftarrow$ info content of v_i
  $\mathcal{E} \leftarrow \{e_{ij} \mid$ weight $w_{ij} \leftarrow$ info overlap$(v_i, v_j)\}$
end
$\mathcal{S} \leftarrow \phi$;
repeat
  forall $v_i \in \mathcal{V}$ **do**
    $N_i \leftarrow$ neighbours of v_i;
    $G_{v_i} \leftarrow r_i + \sum_{v_j \in N_i} r_j \times w_{ij}$;
  end
  $v_{\mathcal{I}} \leftarrow$ node with maximum G_{v_i} ;
  $\Gamma \leftarrow$ subgraph of $\mathcal{G}$ with $v_{\mathcal{I}}$ and its 2-hop neighbourhood ;
  $S_{\mathcal{K}} \leftarrow$ top-$\mathcal{K}$ sentences after sentence compression on $v_i \in \Gamma$;
  $\mathcal{S} \leftarrow \mathcal{S} \cup S_{\mathcal{K}}$;
  forall $v_k \in \mathcal{V}$ **do**
    $r_k \leftarrow r_i \times (1 - w_{\mathcal{S}_{\mathcal{K}} k})$;
  end
until $\forall v_i$, $r_{v_i} \leq$ *threshold*;
C $\leftarrow$ ILP($\mathcal{S}$) ; // Integer Linear Program (Eqns. 2-7)
return *C*

as the 'author' content snippet and the set of documents as the input. Further, we considered the reference summaries as the articles that the author has constructed and is used as a basis for our quality evaluations. The average length of the reference articles is used as the desired length of the generated content.

In order to evaluate the generated content, we use the following **metrics** to measure different aspects of the generated content.

Rouge-N (Lin, 2004) scores measure the precision, recall and f-measure for the occurrence of n-grams in the generated summary with respect to the reference human generated summary. We compute the Rouge scores of the generated content against the reference article.

To compute the **coherence** of the generated content, we follow the definition of local coherence by (Lapata and Barzilay, 2005). Measuring local coherence is quantified by the the degree of semantic relatedness between consecutive sentences and is obtained by taking the mean of all individual transitions. Coherence is given by,

$$coherence(c_{\text{gen}}) = \frac{\sum_{i=1}^{n-1} sim(c_{\text{gen}}^{i}, c_{\text{gen}}^{i+1})}{n-1} \tag{8}$$

where $sim(c_{\text{gen}}^{i}, c_{\text{gen}}^{i+1})$ is a measure of semantic similarity between sentences c_{gen}^{i} and c_{gen}^{i+1}.

To measure the **relevance** of the generated content to the input snippet, we follow the formulation by (Srinivasan et al., 2017). Relevance is computed as a weighted average given by,

$$rel(c_{\text{ref}}, c_{\text{gen}}) = \sum_{i=1}^{M} \frac{\sum_{k=1}^{TopK(c_{\text{ref}}, c_{\text{gen}}^{i})} \gamma^{k} sim(c_{\text{gen}}^{i}, c_{\text{ref}}^{k})}{M \sum_{k=1}^{K} \gamma^{k}}, \tag{9}$$

where c_{ref} are the sentences in the reference and $c_{\text{gen}}^{1 \ldots M}$ are the sentences in the generated content. $TopK(c_{\text{ref}}, c_{\text{gen}}^{i})$ returns the top-K sentences in the reference content similar to c_{gen}^{i} (in the decreasing order of their similarly as computed by $sim(c_{\text{exp}}^{i}, c_{\text{inp}}^{k})$). The parameter γ, $(0 \leq \gamma \leq 1)$, penalizes the addition of a sentence that is similar to only a small set of the input sentence via a decayed-weighted-average.

Finally, to calculate the **information diversity** in the generated content, we again extend the formula-

tion by (Srinivasan et al., 2017) and compute a weighted average given by,

$$div(c_{\text{gen}}) = \sum_{i=1}^{N} \frac{\sum_{k=1}^{TopK(c_{\text{gen}}, c^i_{\text{gen}})} \gamma^k sim(c^i_{\text{gen}}, c^k_{\text{gen}})}{N \sum_{k=1}^{K} \gamma^k}. \quad (10)$$

4.1 Information Coverage and Diversity:

Picking content from a corpus to construct a text can yield redundant content because of multiple representations of the same information in the corpus. If the final content is budgeted, this can result not only in redundant content, but also in a lost opportunity to cover other aspects of the content. In order to address this, we extend the sentence compression (Filippova, 2010) to combine multiple such sentences to provide a compressed content.

However, performing the sentence compression on a larger set of sentences may result in the compression of totally different pieces of information and can often be detrimental to information quality. Our proposed graph formulation identifies similar sentences to be compressed. The iterative subgraph selection in our formulation also enables us to cover different aspects of the content that are required to be generated.

Fig. 1(a) shows the performance of the sentence-compressed content against the uncompressed content selected based on relevance maximization (Carbonell and Goldstein, 1998). We have used box-and-whiskers plots for our results in this paper. The box gives the boundaries of the first and third quartile, and the middle band shows the median value. The whiskers extending from the box boundaries on both sides show the minimum and maximum values, except for the outliers which are shown as points by themselves.

The Rouge scores are comparable across the 3 methods indicating similar quality of the different settings. Similarly, the relevance of the generated content is also comparable across the 3 methods. Since the graph based compression iteratively selects candidates to compress based on those previously generated, it produces content that is more diverse as indicated in Fig. 1(a). Since the plain sentence compression works on a large set of candidates, the diversity improvement is lower than that for the graph based compression. This validates our hypothesis that graph based compression yields diverse yet relevant content.

Finally, the graph based sentence compression improves the overall readability of the text based on the Flesch reading ease (Kincaid et al., 1975) suggesting that the improvements observed with relevance and diversity are not coming at the cost of the linguistic quality of the generated content.

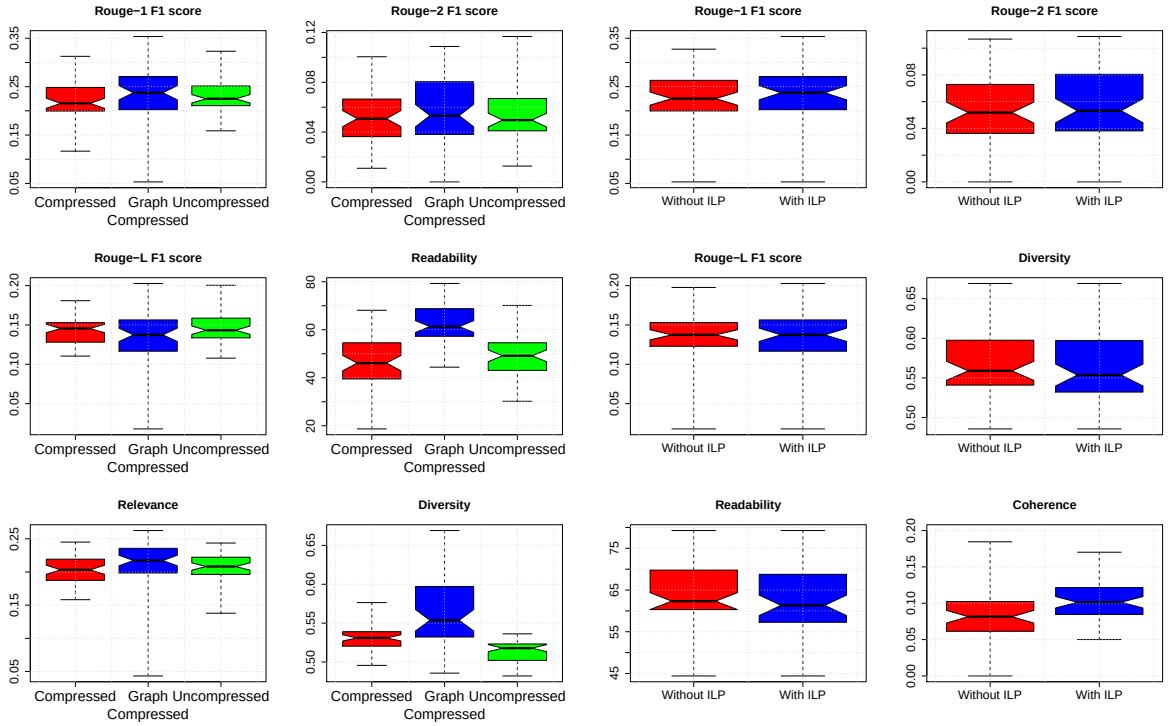
4.2 Coherence with ILP formulation:

The ILP formulation reorders the generated content in a way that improves the overall coherence of the content. Fig. 1(b) shows the performance of the generation with and without ILP. For the non-ILP version, candidates were selected from the compressed sentences via relevance maximization (Carbonell and Goldstein, 1998). It can be seen that the ILP formulation improves the overall coherence of the generated content while maintaining the other quality aspects.

5 Performance evaluation of CCC against baselines

To position our work against existing work, we use two state-of-the-art baseline systems. Our first baseline is GEMS, proposed by (Taneva and Weikum, 2013). GEMS is an extractive generation framework that uses an ILP formulation to extract a set of sentences towards the required content optimizing for the coherence by selecting contiguous sentence fragments. Our second baseline, WikiWrite (Banerjee and Mitra, 2016), has been used to generate Wikipedia articles in a coherent fashion. Similar to our approach, WikiWrite also uses a combination of sentence compression with ILP, however, the sentence compression does not account for the information covered by previously generated candidates, leading to lesser diversity.

Our goal is to obtain the best generation given a set of retrieved text. Alternative methods to modify *tf*idf* criteria (Lecorvé et al., 2008) could be used to increase the specificity of retrieved documents to



(a) Information Coverage & Diversity in the generated content (b) Coherence of the generated content with and without ILP with and without Sentence Compression

Figure 1: Impact of different stages of CCC on the various quality aspects of the generated content

the topics in the query. However, to prevent any bias in the quality of query search results, we ensure that the seed set of documents provided as input to each of the approaches is the same (specified in Section 3) without any such modifications.

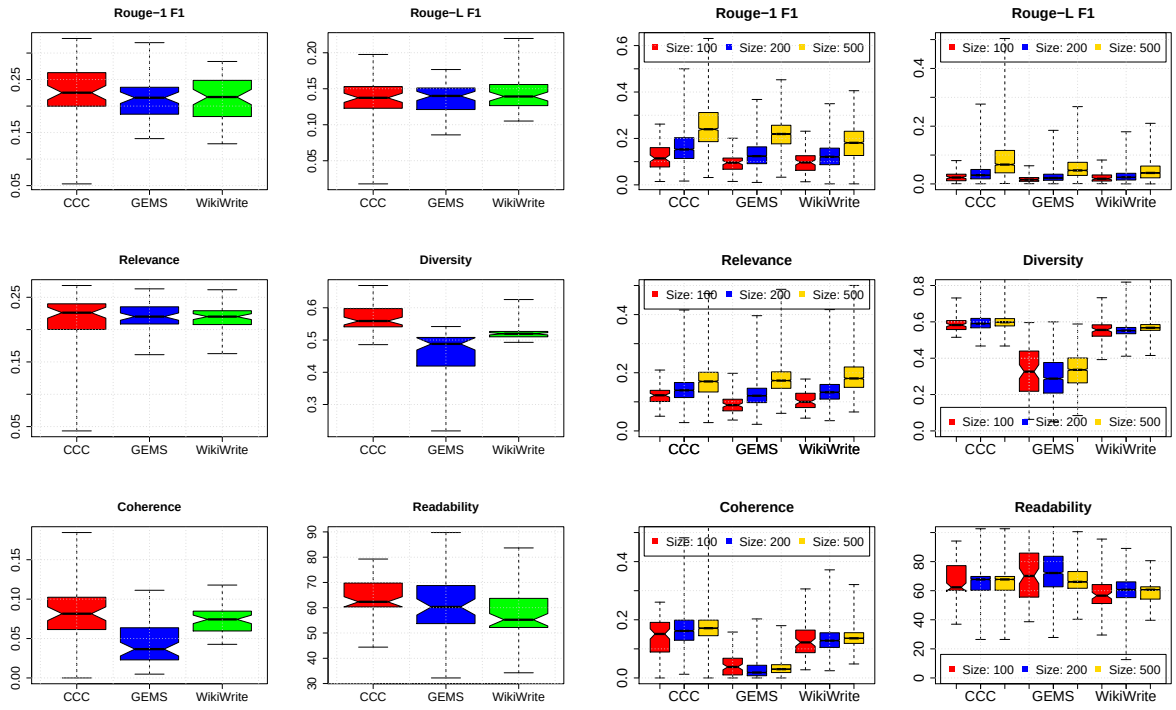
In the rest of this section, we present a comparison of CCC against the baselines. To check for statistical significance, we employ the Wilcoxon signed-ranked sum test (Wilcoxon, 1945), comparing the paired difference in values between the baseline and CCC. Test statistic and p-value indicated are from one sided tests, as indicated. The significance level is taken at $p = 0.05$. We also report the sum of ranks assigned to differences with positive sign (V) for each test.

Fig. 2(a) shows the Rouge scores for the generated content against the human generated content. At the entity level, the proposed generation has a significantly better performance against both the extractive (GEMS) ($V = 977$, $p = 0.042$) and abstractive (WikiWrite) ($V = 979, p = 0.040$) approaches. When compared beyond unigrams, the generation quality is comparable as indicated by the Rouge-L scores.

Since all the methods optimize for the relevance, the proposed approach has comparable relevance to the baselines. However, our proposed approach yields a higher information diversity and coverage as compared to the other approaches. As observed previously, sentence compression results in more information compressed into the content and hence both WikiWrite and our approach generate more diverse content. The diversity of our approach is more than that of WikiWrite due to the optimized sentence selection that precedes compression in our approach. This indicates that our approach yields an article much **more informative** than those generated by the baselines - GEMS ($V = 1540$, $p < 0.001$) and WikiWrite ($V = 1440$, $p < 0.001$).

GEMS is outperformed by both our approach ($V = 1440, p < 0.001$) and Wikiwrite ($V = 1398$, $p < 0.001$) at coherence, since both of these algorithms account for coherence explicitly. The coherence of our algorithm and Wikiwrite was found to be comparable.

Finally, we compute the Flesch reading ease (Kincaid et al., 1975) to quantify the readability of the generated content as its based on the Flesch reading ease. Since GEMS uses human generated sentences as-is from the input, it yields readable sentences. Unlike WikiWrite, our approach compresses only



(a) DUC 2007 Data

(b) Australian Legal Data

Figure 2: Generated content quality for CCC against GEMS ((Taneva and Weikum, 2013)) and WikiWrite ((Banerjee and Mitra, 2016)) on the Australian Legal Dataset (Galgani et al., 2012)

related sentences and thus does not result in unusually larger and less readable sentences, thus, our readability is higher ($V = 1380$, $p < 0.001$). The reading ease is comparable to GEMS despite not using the sentences as-is.

To conclude, we are able to achieve a performance which parallels that of the better of the two baselines in most aspects, while significantly outperforming them in other aspects.

Australian Legal Dataset: Although the performance of our proposed algorithm on the DUC2007 dataset is promising, the dataset is relatively small as compared to an enterprise content repository. In order to test the performance on a larger enterprise-like corpus, we evaluate the proposed approach on the Australian Legal Case Reports dataset[1] (Galgani et al., 2012), a collection of 3890 legal cases from the Federal Court of Australia (FCA) from 2006 to 2009. We used the gold standard summary for every case as the input snippet and the actual article as the reference. The legal articles were 6406-word long on an average, while the summaries were 65-word long on an average. Here, we also try to generate articles of different length - 100, 200 and 500 - to test the robustness of extraction. Fig. 2(b) shows the generation quality for our approach against the baselines. The trends observed in Fig. 2(a) for the DUC dataset carry over to Australian Legal Case Reports one as well. This validates the performance of our approach across corpora.

As the target size of generated content is increased, the algorithms' output becomes increasingly more relevant to the original article. The diversity of the proposed algorithm also increases with a minor dip in coherence. This can be attributed to the fact that the coherence measures inter-sentence topical overlap and with increased diversity, there is a minor dip in local coherence since the topic overlap will reduce.

[1] `http://archive.ics.uci.edu/ml/datasets/Legal+Case+Reports`, Accessed 16 March 2017.

6 Conclusion

We studied the problem of constructing a piece of text by picking content from an existing corpus and combining them by optimizing relevance, information coverage and coherence with reduced redundancy. Evaluations based on several metrics against state-of-the-art baselines shows superior performance of the proposed solution and hence demonstrates the feasibility of the approach. We believe that automated text generation will play a key role in smart authoring workflows for several domains in the near future.

References

Anne Aula. 2003. Query formulation in web information search. In *ICWI*.

Ramakrishna B. Bairi, Raghavendra Udupa, and Ganesh Ramakrishnan. 2016. A framework for task-specific short document expansion. In *ACM International Conference on Information and Knowledge Management (CIKM)*.

Siddhartha Banerjee and Prasenjit Mitra. 2016. Wikiwrite: Generating wikipedia articles automatically. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Siddhartha Banerjee, Prasenjit Mitra, and Kazunari Sugiyama. 2015. Multi-document abstractive summarization using ilp based multi-sentence compression. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Lidong Bing, Piji Li, Yi Liao, Wai Lam, Weiwei Guo, and Rebecca Passonneau. 2015. Abstractive multi-document summarization via phrase selection and merging. In *53rd Annual Meeting of the Association for Computational Linguistics (ACL)*.

Jaime Carbonell and Jade Goldstein. 1998. The use of MMR, diversity-based reranking for reordering documents and producing summaries. In *21st ACM SIGIR Conference on Research and Development in Information Retrieval*.

Hoa T. Dang. 2007. Overview of DUC 2007. In *Document Understanding Conference*.

Pawan Deshpande, Regina Barzilay, and David R Karger. 2007. Randomized decoding for selection-and-ordering problems. In *North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*.

Katja Filippova. 2010. Multi-sentence compression: Finding shortest paths in word graphs. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING)*.

Filippo Galgani, Paul Compton, and Achim Hoffmann. 2012. Combining different summarization techniques for legal text. In *Workshop on Innovative Hybrid Approaches to the Processing of Textual Data*.

J Peter Kincaid, Robert P Fishburne Jr, Richard L Rogers, and Brad S Chissom. 1975. Derivation of new readability formulas (automated readability index, fog count and flesch reading ease formula) for navy enlisted personnel. Technical report, Naval Technical Training Command Millington TN Research Branch.

Mirella Lapata and Regina Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. In *International Joint Conference on Artificial Intelligence (IJCAI)*.

Gwénolé Lecorvé, Guillaume Gravier, and Pascale Sébillot. 2008. An unsupervised web-based topic language model adaptation method. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Yanran Li and Sujian Li. 2014. Query-focused multi-document summarization: Combining a topic model with graph-based semi-supervised learning. In *23rd International Conference on Computational Linguistics (COLING)*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out: ACL 2004 workshop*.

Rada Mihalcea and Andras Csomai. 2007. Wikify!: Linking documents to encyclopedic knowledge. In *ACM Conference on Information & Knowledge Management (CIKM)*.

Mir Tafseer Nayeem and Yllias Chali. 2017. Extract with order for coherent multi-document summarization. In *TextGraphs-11: Workshop on Graph-based Methods for Natural Language Processing*.

Ani Nenkova and Kathleen McKeown. 2011. Automatic summarization. *Springer Information Retrieval*.

Daraksha Parveen, Mohsen Mesgar, and Michael Strube. 2016. Generating coherent summaries of scientific articles using coherence patterns. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Christina Sauper and Regina Barzilay. 2009. Automatically generating wikipedia articles: A structure-aware approach. In *47th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Nico Schlaefer, Jennifer Chu-Carroll, Eric Nyberg, James Fan, Wlodek Zadrozny, and David Ferrucci. 2011. Statistical source expansion for question answering. In *ACM International Conference on Information & Knowledge Management (CIKM)*.

Balaji Vasan Srinivasan, Rishiraj Saha Roy, Harsh Jhamtani, Natwar Modani, and Niyati Chhaya. 2017. Corpus-based automatic text expansion. *18th International Conference on Computational Linguistics and Intelligent Text Processing*.

Bilyana Taneva and Gerhard Weikum. 2013. Gem-based entity-knowledge maintenance. In *22nd ACM International Conference on Information & Knowledge Management (CIKM)*.

Lu Wang, Hema Raghavan, Vittorio Castelli, Radu Florian, and Claire Cardie. 2013. A sentence compression based framework to query-focused multi-document summarization. In *51st Annual Meeting of the Association for Computational Linguistics (ACL)*.

Xun Wang, Masaaki Nishino, Tsutomu Hirao, Katsuhito Sudoh, and Masaaki Nagata. 2016. Exploring text links for coherent multi-document summarization. In *The 26th International Conference on Computational Linguistics (COLING)*.

Frank Wilcoxon. 1945. Individual comparisons by ranking methods. *Biometrics bulletin*.