Extract with Order for Coherent Multi-Document Summarization

Mir Tafseer Nayeem, Yllias Chali

University of Lethbridge

Lethbridge, AB, Canada

TextGraph 2017

What is summarization?

- The process of finding the **most relevant informations** in a text and presenting them in a **condensed** form.
- Single Document Summarization
  - Given a single document produces abstract, outline or headline
- **Multi-Document Summarization**
  - A cluster of related documents about the same topic

- Summaries can be classified as:
  - **Extractive**
    - Extract important sentences from the original text without any modification.
  - Abstractive
    - Abstractive methods rewrite sentences from scratch, involving compression, fusion and paraphrasing.

Related Research Works

- Early Works:
  - **Graph-based** methods for computing sentence importance.
    - LexRank (Erkan and Radev, 2004) and TextRank (Mihalcea and Tarau, 2004)
  - **Supervised model** for predicting word importance.
    - RegSum system (Hong and Nenkova, 2014)
  - Summarization as a submodular maximization problem (Lin and Bilmes, 2011)
  - All the above systems don't care about the sentence ordering in the output summary.
- Recent Works:
  - **Single document summarization** systems, where sentences are implicitly ordered according to the **sentence position**.
  - Attentional encoder-decoder (Cheng and Lapata, 2016)
  - RNN based sequence classifier (Nallapati et al., 2017)

Contributions

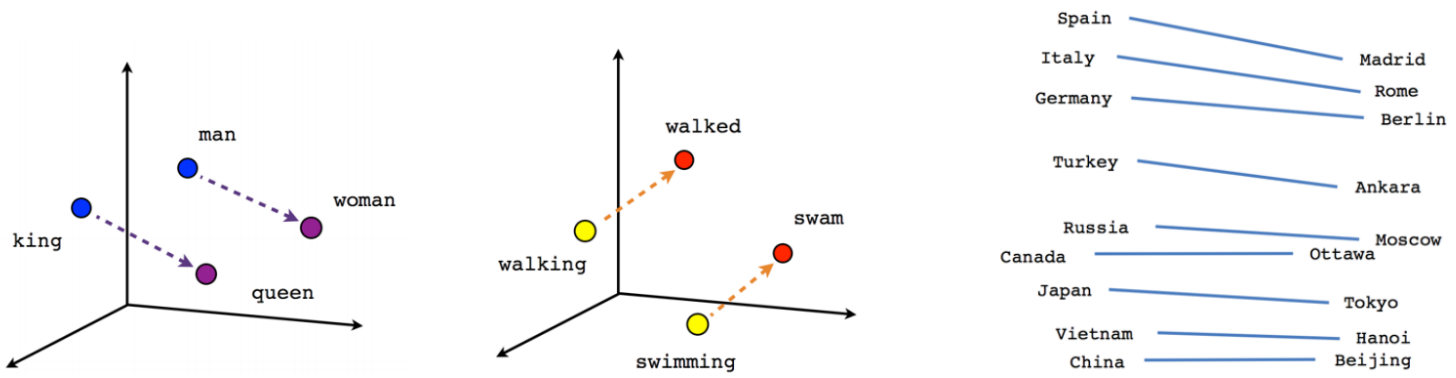
- We implemented an **ILP** (Integer Linear Programming) based sentence selection along with **TextRank** (Mihalcea and Tarau, 2004) scores and **key phrases** for **extractive multi-document** summarization.
- We further model the **coherence** using a greedy algorithm to increase the **readability** of the generated summary.
- We conduct experiments on the Document Understanding Conference (DUC) 2004 datasets using **ROUGE** toolkit.
- Our system achieves significant improvements in terms of **information coverage** and **coherence**.

Outline

- Sentence Similarity
- How to find optimal λ ?
- Sentence Ranking
- Sentence Clustering
- Sentence Selection
- Sentence Ordering
- Sample Generated Summary
- Evaluation
- Baseline Systems & Results
- Limitations & Future Work

Sentence Similarity

- We use **Word2Vec** (Mikolov et al., 2013) which embeds words in a continuous vector space where semantically similar words are placed to nearby points to each other.



- It's a popular method used in many natural language processing applications.
- We use the pre-trained word embedding collected from (Mikolov et al., 2013) to represent a sentence.

Sentence Similarity

- Weighted vector sum according to the term-frequency (**TF**) of a word (w) in a sentence (S).
- E is the **word embedding model** (Mikolov et al., 2013) and $idx(w)$ is the index of the word w.

$$S = \sum_{w \in S} TF(w, S) \cdot E[idx(w)]$$

$$Sim(S_i, S_j) = \lambda \cdot \boxed{NESim(S_i, S_j)} + (1 - \lambda) \cdot \boxed{CosSim(S_i, S_j)}$$

Entity Overlap between sentences

Cosine Similarity between sentence vectors

How to find optimal λ ?

- We use the **SICK** dataset of SemEval-2014 (Marelli et al., 2014) which consists of about 10,000 English sentence pairs with a **relatedness score** [1, 5].
- Pair of sentences with relatedness scores lower than 2 are assumed dissimilar, and the scores higher than 4 are considered similar.
- Other partially related sentences are filtered out.
- The remaining dataset consists of 923 dissimilar sentence pairs and 3305 similar sentence pairs.

λ	**P**	**R**	**F**
0.1	0.88	0.94	0.91
0.2	0.88	0.94	0.91
0.3	**0.89**	**0.95**	**0.92**
0.4	0.86	0.92	0.89
0.5	0.82	0.88	0.85

Sentence Ranking

- We rank the sentences using **TextRank** algorithm (Mihalcea and Tarau, 2004).
- An **undirected graph** is constructed where sentences are vertices, and edge weights are the similarity between vertices (sentences).
- Instead of **lexical overlap**, we use the semantic similarity $Sim(S_i, S_j)$ to form a weighted edge between two sentences.
- After constructing the graph, we can run the **TextRank** algorithm on it by repeatedly applying the following TextRank update rule until convergence.

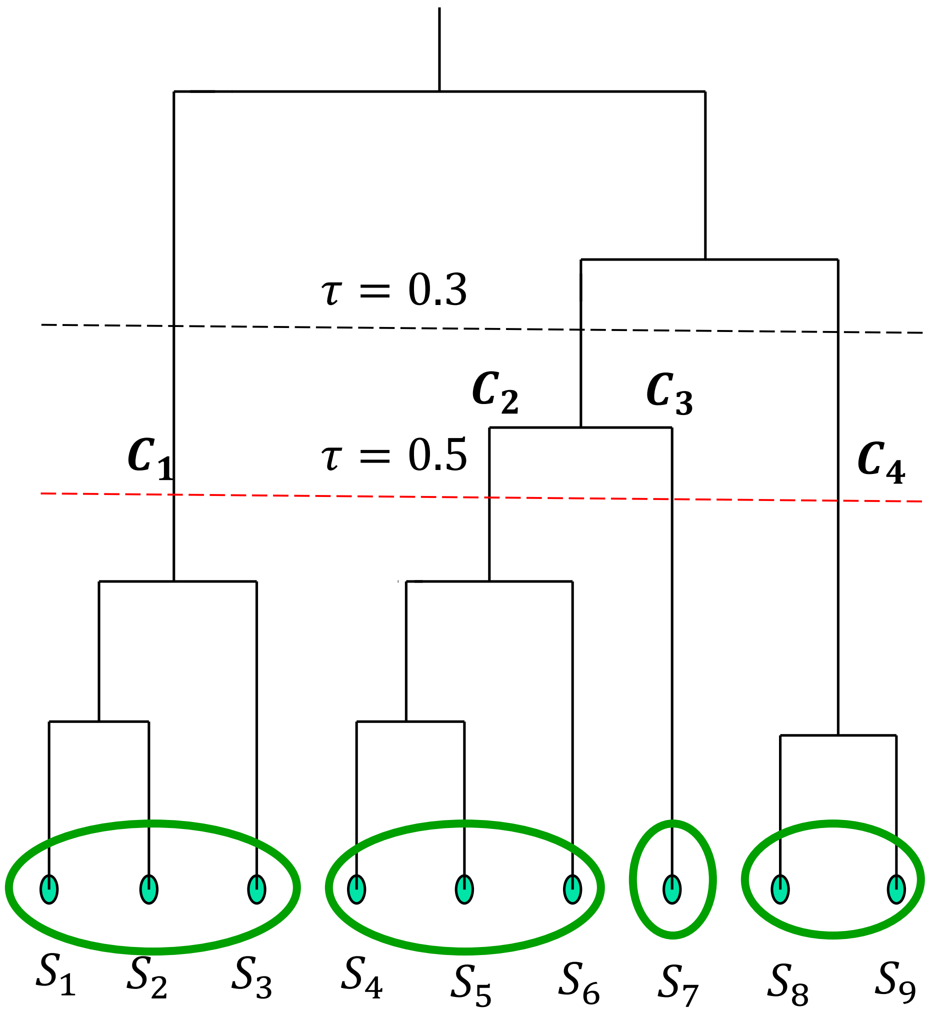
$$Rank(S_i) = (1-d) + d * \sum_{S_j \in N(S_i)} \frac{Sim(S_i, S_j)}{\sum_{S_k \in N(S_j)} Sim(S_j, S_k)} Rank(S_j)$$

- Where $Rank(S_i)$ is the importance score assigned to sentence (S_i), d is the dampening factor which is set to 0.85 as original literature.

Sentence Clustering

- This step is very important for two main reasons.
  - Selecting at most one sentence from each cluster will **decrease redundancy** from the **summary side**.
  - Selecting sentences from the different set of clusters will increase the **information coverage** from the **document side** as well.
- For grouping similar sentences. We use a **hierarchical agglomerative clustering** (Murtagh and Legendre, 2014) with a **complete linkage criteria**.
- In computing the clusters, we use the similarity function $Sim(S_i, S_j)$.
- We set a similarity threshold (τ = 0.5) to stop the clustering process.

Sentence Clustering Process



Sentence Selection

- We use the **concept-based ILP framework** (Gillick and Favre, 2009) with suitable changes to select the best subset of sentences.
- The system **extracts** sentences that cover **important concepts** while ensuring the **summary length** is within a limit.
- Instead of bigrams we use **keyphrases** as concept.
- We extracted keyphrases using **RAKE** tool (Rose et al., 2010). We assign a weight to each keyphrase using the score returned by RAKE.
- In order to ensure only **one sentence per cluster** we add an extra constraint.

Sentence Selection

$$Maximize: (\boxed{\sum_i \overline{w}_i k_i} + \sum_j Rank(S_j) \cdot s_j)$$

Maximize the sum of keyphrase weights

$$Subject\ to: \sum_j l_j s_j \leq L$$

$$s_j Occ_{ij} \leq k_i, \quad \forall i, j$$

$$\sum_j s_j Occ_{ij} \geq k_i, \quad \forall i$$

$$\sum_{j \in g_c} s_j \leq 1, \quad \forall g_c$$

Sentence Selection

$$Maximize:(\sum_i \overline{w}_i k_i + \boxed{\sum_j Rank(S_j)} \cdot s_j)$$

$$Subject\ to:\sum_j l_j s_j \leq L$$

$$s_j Occ_{ij} \leq k_i, \quad \forall i,j$$

$$\sum_j s_j Occ_{ij} \geq k_i, \quad \forall i$$

$$\sum_{j \in g_c} s_j \leq 1, \quad \forall g_c$$

Maximize the sum of sentence rank scores

Sentence Selection

$$Maximize: (\sum_i \overline{w}_i k_i + \sum_j Rank(S_j) \cdot s_j)$$

$$Subject\ to: \boxed{\sum_j l_j s_j \leq L}$$

$$s_j Occ_{ij} \leq k_i, \quad \forall i, j$$

$$\sum_j s_j Occ_{ij} \geq k_i, \quad \forall i$$

$$\sum_{j \in g_c} s_j \leq 1, \quad \forall g_c$$

Summary Length under a certain limit

Sentence Selection

$$Maximize : (\sum_i \overline{w}_i k_i + \sum_j Rank(S_j) \cdot s_j)$$

$$Subject\ to : \sum_j l_j s_j \leq L$$

$$s_j Occ_{ij} \leq k_i, \quad \forall i, j$$

$$\sum_j s_j Occ_{ij} \geq k_i, \quad \forall i$$

Avoiding the repetition of keyphrases

$$\sum_{j \in g_c} s_j \leq 1, \quad \forall g_c$$

Sentence Selection

$$Maximize: (\sum_i \overline{w}_i k_i + \sum_j Rank(S_j) \cdot s_j)$$

$$Subject\ to: \sum_j l_j s_j \leq L$$

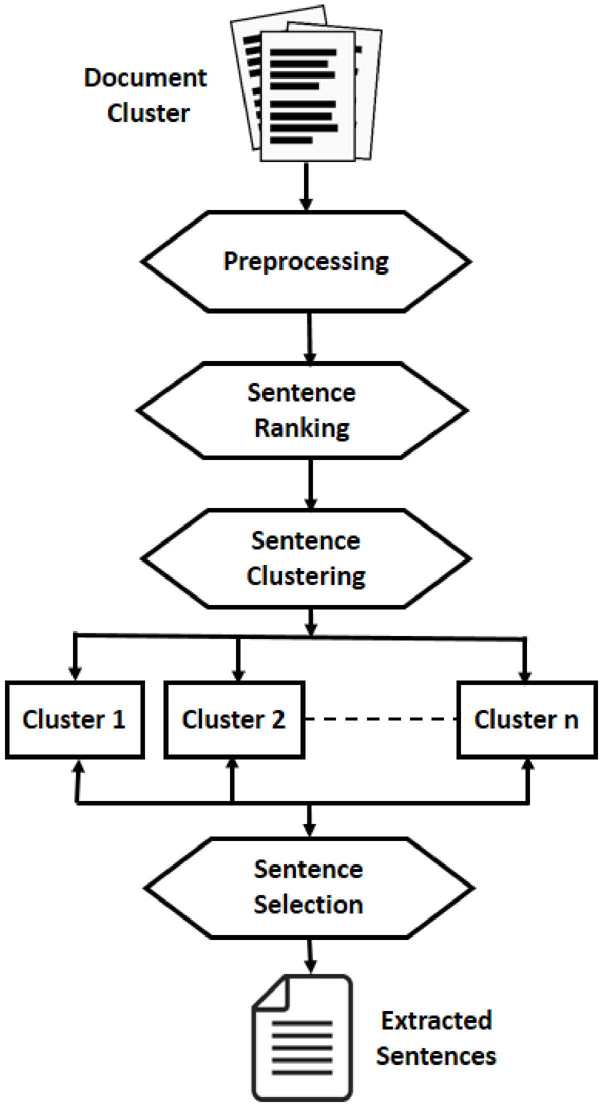
$$s_j Occ_{ij} \leq k_i, \quad \forall i, j$$

$$\sum_j s_j Occ_{ij} \geq k_i, \quad \forall i$$

$$\boxed{\sum_{j \in g_c} s_j \leq 1, \quad \forall g_c}$$

Selects at most one sentence from each cluster

Sentence Extraction Process



Sentence Ordering

- A wrong order of sentences convey entirely different idea to the reader of the summary and make it difficult to understand.
- For single document, summary can be presented by preserving the sentence position in the original document.
- Sentence position does not provide clue to the sentence arrangement in multi-document setting.
- We define coherence as the similarity between all adjacent sentences in a document D.

$$Coherence(D) = \frac{\sum_{i=1}^{n-1} Sim(S_i\ , S_{i+1})}{n-1}$$

Sentence Ordering Algorithm

Algorithm 1: Place a sentence in a document

```
Procedure SentencePositioning(D, S_n)
    Data: Input document D which is assumed sorted. New sentence S_n which we will place
          in the document D.
    Result: Return new document D_n after placing the sentence S_n.
    t ← 1;
    Coh_max ← 0 ;
    D_tmp ← D ;
    l ← DocLength(D) ;
    while t ≤ l + 1 do
        ⇒Place the S_n in t^th position of D_tmp ;
        Coh_tmp ← Coherence(D_tmp);
        if Coh_tmp > Coh_max then
            D_n ← D_tmp;
            Coh_max ← Coh_tmp;
            ⇒ Remove S_n from the t^th position of the document D_tmp ;
        end
        t ← t + 1;
    end
    return D_n;
```

Sample Generated Summary for document set (e.g. d30015t) from DUC-2004 dataset

Summary Generated (After Sentence Extraction)
But U.S. special envoy Richard Holbrooke said the situation in the southern Serbian province was as bad now as two weeks ago. A Western diplomat said up to 120 Yugoslav army armored vehicles, including tanks, have been pulled out. On Sunday, Milosevic met with Russian Foreign Minister Igor Ivanov and Defense Minister Igor Sergeyev, Serbian President Milan Milutinovic and Yugoslavia's top defense officials. To avoid such an attack, Yugoslavia must end the hostilities, withdraw army and security forces, take urgent measures to overcome the humanitarian crisis, ensure that refugees can return home and take part in peace talks, he said.
Summary Generated (After Sentence Ordering)
On Sunday, Milosevic met with Russian Foreign Minister Igor Ivanov and Defense Minister Igor Sergeyev, Serbian President Milan Milutinovic and Yugoslavia's top defense officials. But U.S. special envoy Richard Holbrooke said the situation in the southern Serbian province was as bad now as two weeks ago. A Western diplomat said up to 120 Yugoslav army armored vehicles, including tanks, have been pulled out. To avoid such an attack, Yugoslavia must end the hostilities, withdraw army and security forces, take urgent measures to overcome the humanitarian crisis, ensure that refugees can return home and take part in peace talks, he said.

Evaluation

- Our system **ILPRankSumm** (**ILP** based sentence selection with Text**Rank** for Extractive **Summ**arization)
- Evaluation metric: **ROUGE** Toolkit (Lin,2004)
  - **R-1** (unigram matches)
  - **R-2** (bigram matches)
  - **R-SU4** (skip-bigrams four unigrams in between)
- Dataset : **DUC 2004** (Task-2, Length limit(L) = 100 words)
- We report the **limited length recall** scores for the evaluation metrics.
- ROUGE scores can not determine the **summary coherence**.
- We evaluate summary coherence using (Lapata and Barzilay, 2005) (Barzilay and Lapata, 2008) which output **coherence probabilities** for an ordered set of sentences.

Baseline Systems & Results

- Baseline Systems
  - **LexRank** (Erkan and Radev, 2004)
  - **GreedyKL** (Haghighi and Vanderwende, 2009)
- State-of-the-art Systems
  - **Submodular** (Lin and Bilmes, 2011)
  - **ICSISumm** (Gillick and Favre, 2009)
- The summaries generated by the above extractive summarizers were collected from (Hong et al., 2014)

System	Models	R-1	R-2	R-SU4	Coherence
Baseline	LexRank	35.95	7.47	12.48	0.39
	GreedyKL	37.98	8.53	13.25	0.46
State-of-the-art	Submodular	39.18	9.35	**14.22**	0.51
	ICSISumm	38.41	9.78	13.31	0.44
Proposed System	ILPRankSumm	**39.45**	**10.12**	14.09	**0.68**

Limitations & Future Work

- According to (Hong et al., 2014) all the summarizer from the previous research either truncated the summary to **100th word**, or removed the **last sentence** from the summary set.
- First method produces a certain **ungrammatical sentence**.
- Second one may **lose a lot of information** in the worst case, if the sentences are long.
- In this paper, we follow the second one to produce **grammatical summary** .
- In future, we will propose a solution for the **length limit problem**.

Thank You! ☺
Questions?

mir.nayeem@uleth.ca