SYSTEMS ENGINEERING RESEARCH CENTER

SERC SPONSOR 12TH ANNUAL RESEARCH REVIEW

NOVEMBER 18, 2020

STEVENS INSTITUTE of TECHNOLOGY
THE INNOVATION UNIVERSITY®

WRT-1017: Keyphrase Extraction using Language Embeddings

Sponsor: US Army CCDC

Dr. K.P. (Suba) Subbalakshmi

12th Annual SERC Sponsor Research Review
HOSTED VIRTUALLY ON: November 18, 2020
www.sercuarc.org

AI Enabled Keyphrase Extraction

- Keyphrase extraction is the first step for a lot of downstream NLP tasks needed for competitive intelligence
  - Select relevant articles for SME attention
  - Rank ideas according to importance
  - Extract trends
  - Discover areas of interest
  - Create summaries of documents

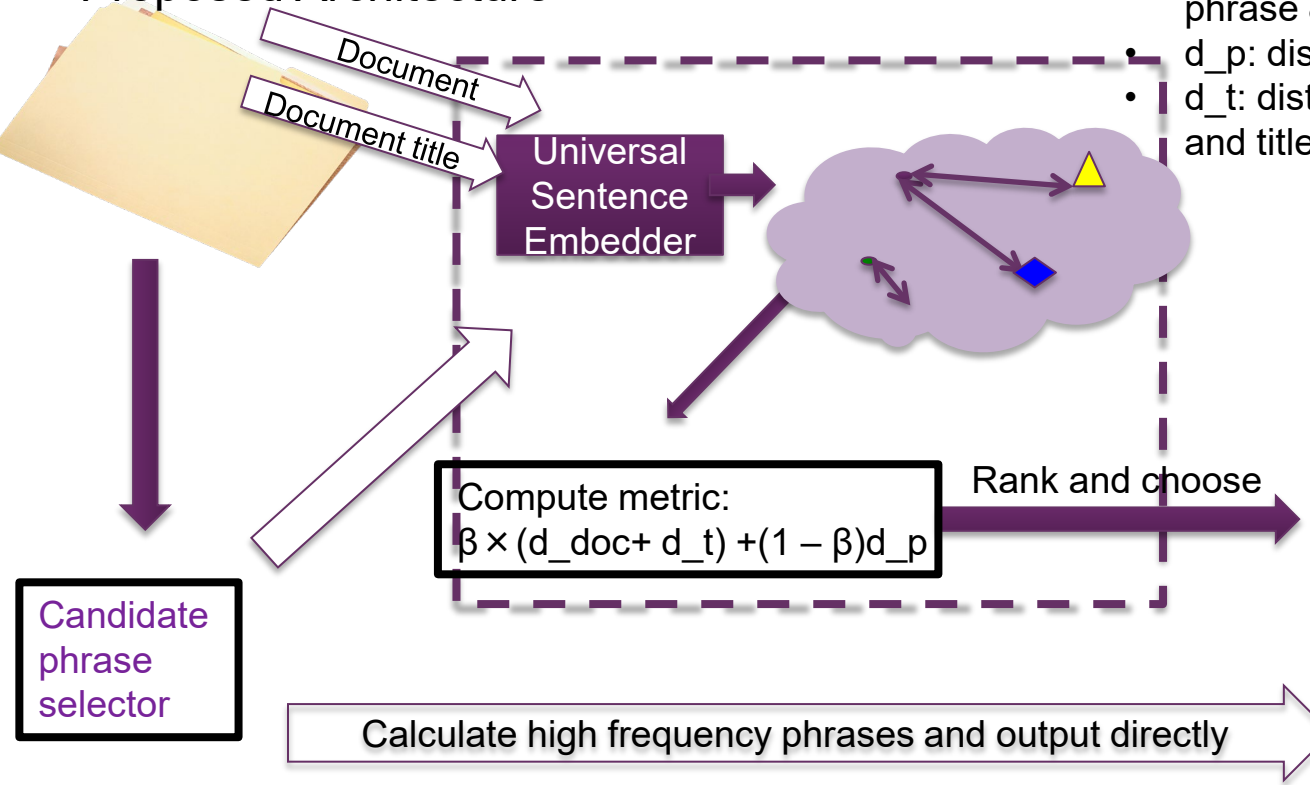
Goal of work

- **Goals:** propose new keyphrase extraction algorithms that meets or exceeds state-of-the-art (SOTA) methods
  - Balance diversity of keyphrases with relevance
  - Use title of article to inform keyphrase extraction
  - Use language embedding to represent keyphrases, documents and title
  - Suggest novel make-up of keyphrases

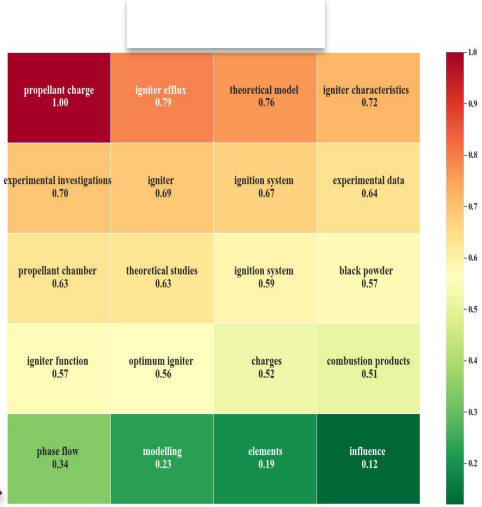
Innovation of our algorithm

- **Title Similarity**:
  - Take advantage of relationship between the article title and keyphrases
- **Frequency**:
  - Keyphrases are likelier to appear more often that other types of words
  - Select these phrases as candidates
- **Candidate keyphrases**:
  - Suggest new structure for keyphrases
- **New Metric (incorporating diversity and title information)**:
  - New metric proposed to trade-off relevance of keyphrases with the similarity *between* keyphrases

Proposed Architecture



- d_doc: distance between candidate phrase & document
- d_p: distance between candidate phrases
- d_t: distance between candidate phrase and title



Proposed Method: Extracting candidate phrases

- Extract candidate phrases from the text
  - Keyphrases, (as opposed to keywords) always consist of two or more words and have some fixed part-of-speech types
- Proposed make-up of candidate keyphrases
  - Adj(one or more) + Nouns (One or more)
  - Nouns (one or more) + Nouns (one or more)
  - VBN + Nouns → example: ``distributed system"
  - Adj + VBG + Nouns → example: ``minimal generating sets"
  - Adj + VBN + Nouns →example: "out-of-print print music"
- *EmbedRank uses Adj (zero or more) + Nouns (One or more)*

First step *(contd.)* : Extract candidate phrases

- Delete unimportant words for candidate phrase
  - **e.g:** `Such More Other Most Less Few Little Many Better`
  - Example of trimmed phrases:
    - ['such information architecture'] → [‘information architecture']
- Compute high frequency keyphrases (that occur two or more times) and output them directly as predicted keyphrases

Third step: Language Embedding

- Use sentence embedding to represent:
  - the candidate phrases
  - the title
  - the document in the same high-dimensional vector space
- We use the Universal Sentence Encoder (USE)
  - Proposed by Google and uses either the transformer architecture or the deep averaging network
  - Considered the SOTA for language embedding
  - Produces a 512 dimension vector for each phrase/document/sentence

- EmbedRank (one of the comparison algorithms) -
  - an unsupervised method to automatically extract keyphrases from a document
  - Uses sent2vec as embedding method for sentences

Sent2Vec

β * sim_to_doc - (1-β)* sim_between.

Extract candidate phrases → Embed candidate phrases and the document → Rank and Select

Step 1 **Step 2** **Step 3**

The Metric

- We propose a new metric to trade-off the relevance of the keyphrase against the similarity between the keyphrases
  - Ensures diversity and relevance of keyphrases
- β* (norm_sim_doc + norm_sim_title) + (1-β)* norm_sim_between
  - **norm_sim_doc** : normalized cosine similarity between the document and each candidate phrase
  - **norm_sim_title** : normalized cosine similarity between the title and each candidate phrase
  - **norm_sim_between**: normalized cosine similarity between candidate phrases
  - **β:** Controls the relative importance between relevance and diversity.

Rank and Select Keyphrases

- The score for each candidate keyphrase is calculated using the proposed metric
- Keyphrases are ranked and the top N keyphrases are selected
- Evaluation Metrics:
  - We evaluate our method against two most recent methods
  - Using two different metrics

Evaluation Method 1:

Precision: # Matching predicted keyphrases / Total # predicted keyphrases

Recall: # Matching predicted Keyphrases / Total # ground-truth keyphrases

F1 score: 2 * (precision * recall) / (precision + recall)

Comparisons

N	Method	Inspec	Nus
5	Embedrank	31.51	10.28
	Multipartite	25.78	13.31
	Ours	**34.18**	**16.32**
10	Embedrank	37.94	13.43
	Multipartite	30.01	16.92
	Ours	**40.64**	**17.86**
15	Embedrank	38.40	14.47
	Multipartite	31.44	17.16
	Ours	**42.29**	**18.67**

Multipartite: The state-of-the-art method for the NUS dataset mentioned in the Embedrank paper.

Evaluation Method 2

- Precision: # Matching predicted keyphrases / total # predicted keyphrases.
- Recall: # Matching predicted keyphrases / total # ground-truth keyphrases present (Ignore the absent keyphrases)
- F1 score: 2 * (precision * recall) / (precision + recall)
  - Metric proposed in: “One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases (ACL2020)”

Evaluation Result: Compare to state-of-the-art method

	Inspec			Krapivin			NUS			SenEval		
Model	F1--5	F1--10	F1--X	F1--5	F1--10	F1--X	F1--5	F1--10	F1--X	F1--5	F1--10	F1--X
Abstractive Neual												
CopyRNN	24.4	28.9	29	30.5	26.6	32.5	37.6	35.2	40.6	31.8	31.8	31.7
KG-KE-KR-M	25.7	28.4	31.4	27.2	25	31.7	28.9	28.6	38.4	20.2	22.3	30.3
CateSeqD	27.6	33.3	33.1	32.5	28.5	37.1	37.4	36.6	40.6	32.7	35.2	35.7
Extractive Neural												
DivGraphPointer	38.6	41.7		46.2	40.2		40.1	38.9		36.3	29.7	
w/Additional Data												
Semi-Multi	32.8	31.8		32.3	25.4		36.5	32.6		31.9	31.2	
TG-Net	31.5	38.1		34.9	29.5		40.6	37		31.8	32.2	
Extractive IR												
Tfidf	16	24.4	20.8	6.7	9.3	6.8	11.2	14	12.2	8.8	14.7	11.3
TextRank	28.6	33.9	33.5	33.5	16	33.5	23	21.6	23.8	21.7	22.6	22.9
KEA	2.2	2.2	2.2	1.8	1.7	1.7	7.3	7.1	8.1	6.8	6.5	6.6
Maui	3.5	4.6	3.9	0.5	0.7	0.6	0.4	0.6	0.6	1.1	1.4	1.1
Ours	39.12	44.51	45.25	25.78	23.45	26.8	29.48	27.43	30.09	25.52	27.81	28.92

Here X denotes the number of ground-truth keyphrases present in the original document.

Conclusion

- INSPEC: Our method is better than all models: supervised methods and unsupervised methods.

- NUS, Krapivin and SemEval : Our method is better than all unsupervised methods. And it is close to some supervised models and better than some supervised methods.

Result analysis

Datasets	INSPEC	Semeval	Karpivin	NUS
number of ground-truth keyphrases	3858	671	1485	1263
number of single word ground-truth keyphrases	531(13.76%)	161(23.99%)	307(20.67%)	409(32.38%)
number of overlapping ground-truth keyphrases	10	4	1	0
Ground-truth keyphrases: ground-truth keyphrases present in the original document.				

Because our method focuses on keyphrases which have two or more words, we get significant improvements over other methods for INSPEC