Addressing Overgeneration Error: An Effective and Efficient Approach to Keyphrase Extraction from Scientific Papers

UNC CHARLOTTE

Haofeng Jia and Erik Saule

Dept. of Computer Science, UNC Charlotte

hjia1@uncc.edu, esaule@uncc.edu

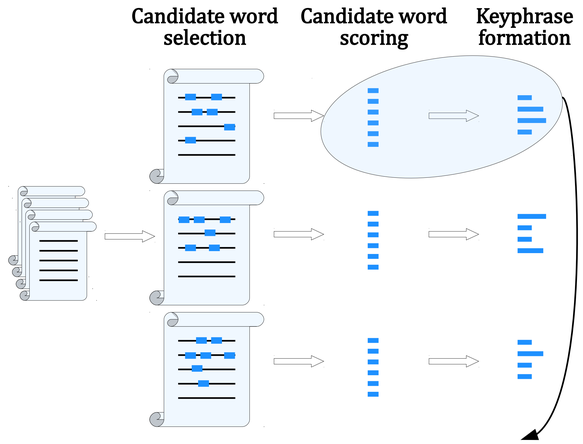
Keyphrase Extraction

Problem Definition: Given a set of scientific documents, extract those words and phrases that provide a brief and precise description for each document.

Challenge

Challenge: Overgeneration errors occur when a system erroneously outputs all candidates phrase as keyphrases because they contain a highly scored word. Current keyphrase extraction systems typically assign scores to words firstly, and rank candidate phrases according to the sum of weights of their component words. Therefore, this kind of mechanism tends to suffer from overgeneration errors.

Example: Overgeneration Error



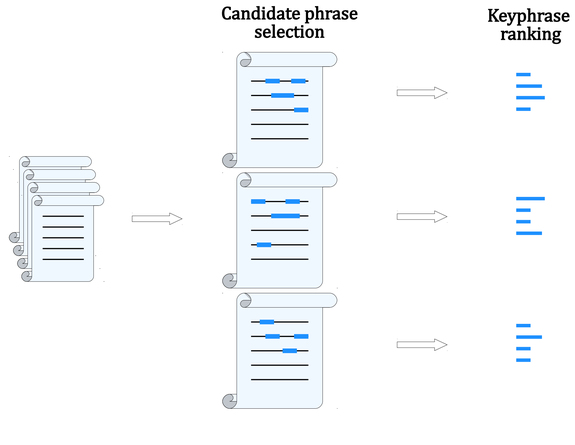
Candidate word scoring
graph 0.00214
structures 0.00162
learning 0.00095
network 0.00084
k-partite 0.00071
model 0.00068
hidden 0.00065
...

Top k	Keyphrase Formation
1	original k-partite graph
2	**k-partite graph**
3	hidden structures
4	various structures
5	local cluster structures
6	global cluster structures
7	**relation summary network**
...	...

KeyPhraser

Idea: In order to alleviate this problem, we look for a way to allow us to directly operates on phrases instead of their component words.

- What kinds of properties make a group of words into a phrase?
- What kinds of properties make a phrase into a keyphrase?
- What is special for scientific documents?



- Concordance: $$Conc(s) = \begin{cases} 1 & \text{if } s = [adj]^*[noun]^+ \\ 0 & \text{otherwise} \end{cases}$$
- Popularity: $$Pop(s,d) = log(f(s,d)+1)$$ where $f(s,d)$ denotes the frequency of a phrase $s \in \mathcal{P}$ in the document d.
- Informativeness: $$Info(s) = log\frac{|\mathcal{C}|}{|d \in \mathcal{D} : s \in d|}$$ where $\mathcal{C}$ means the whole corpus, and $\mathcal{D}$ means the documents that contain candidate phrase s.
- Positional Preference: $$Pos(s,d) = log(\sum_{\text{each s in d}} \frac{|d|}{op(s,d)+1})$$ where $op(v,d)$ denotes an occurrence position of phrase v in document d. An alternative way only takes the first occurrence position of a phrase into consideration. $$Pos(s,d) = log\frac{|d|}{fop(s,d)+1}$$ where $fop(v,d)$ denotes the first occurrence position of phrase v in document d.

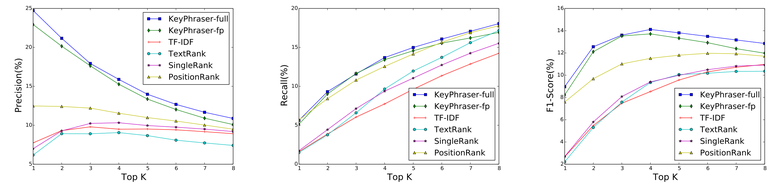
$$Keyphraser(s,d) = Conc(s)Pop(s,d)Info(s)Pos(s,d)$$

Data Sets

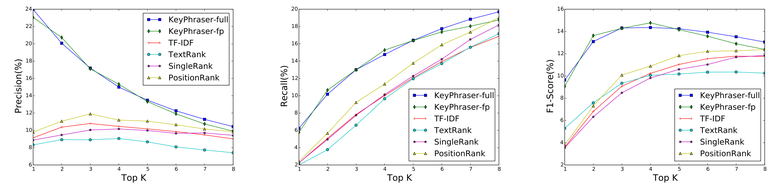
Statistics	WWW	KDD
# of Papers	1331	755
average # of ground truth keyphrases for each paper	4.6	3.8
average # of words in each ground truth keyphrase	1.9	1.8

Experimental Results

Performance: on WWW data set:



Performance: on KDD data set:



Example: Predicted Keyphrases Comparison

k	SingleRank	PositionRank
1	original k-partite graph	original k-partite graph
2	**k-partite graph**	**k-partite graph**
3	hidden structures	various structures
4	various structures	hidden structures
5	local cluster structures	local cluster structures
6	global cluster structures	global cluster structures
7	**relation summary network**	**unsupervised learning**
8	general model	**relation summary network**
k	KeyPhraser-fp	KeyPhraser-full
1	**unsupervised learning**	**k-partite graph**
2	**k-partite graph**	hidden structures
3	hidden structures	**unsupervised learning**
4	data objects	**relation summary network**
5	multiple types	clustering approaches
6	**relation summary network**	data objects
7	general model	multiple types
8	local cluster structures	connections

Conclusion

- KeyPhraser takes phrases instead of component words as semantic unit to address overgeneration errors;
- KeyPhraser achieves better results on precision, recall and F-score;
- KeyPhraser is particularly good at finding more keyphrases at small k, which is essential for real systems;
- KeyPhraser is fast to compute.

References

[1] Mihalcea, R., Tarau, P.: Textrank: Bringing order into texts. In: Proc. of EMNLP. (2004)
[2] Wan, X., Xiao, J.: Single document keyphrase extraction using neighborhood knowledge. In: Proc. of AAAI. (2008)
[3] Hasan, K.S., Ng, V.: Conundrums in unsupervised keyphrase extraction: making sense of the state-of-the-art In: Proc of COLING. (2010)
[4] Hasan, K.S., Ng, V.: Automatic keyphrase extraction: A survey of the state of the art. In: Proc of ACL. (2014)
[5] Florescu, C., Caragea, C.: Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In: Proc of ACL. (2017)

Acknowledgments

This material is based upon work supported by the National Science Foundation under Grant No. 1652442.