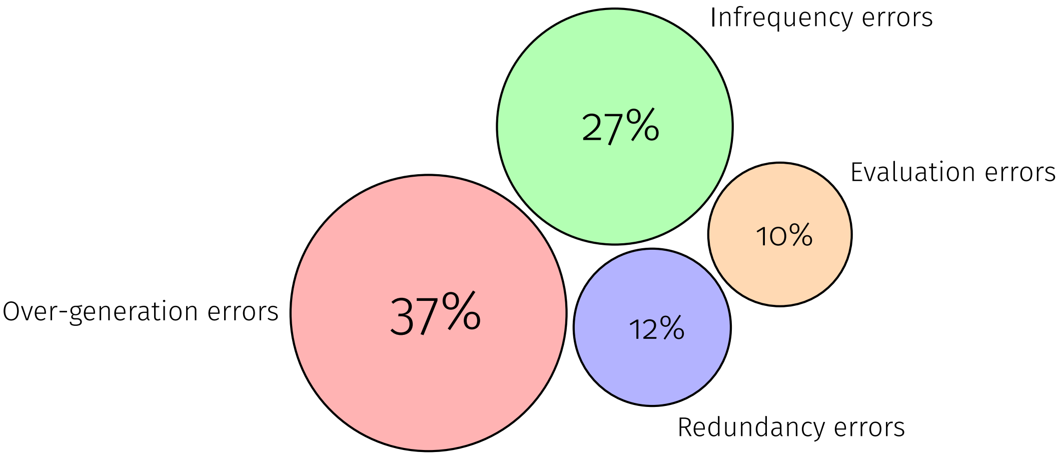
Reducing Over-generation Errors for Automatic Keyphrase Extraction using Integer Linear Programming

Florian Boudin

LINA - UMR CNRS 6241, Université de Nantes, France

Keyphrase 2015

Errors made by keyphrase extraction systems



[Hasan and Ng, 2014]

Motivation

- Most errors are due to over-generation
  - System correctly outputs a keyphrase because it contains an important word, but erroneously predicts other candidates as keyphrases because they contain the same word
  - e.g. **olympics**, **olympic** movement, international **olympic** comittee
- Why over-generation errors are frequent?
  - Candidates are ranked independently, often according to their component words
- We propose a global inference model to tackle the problem of over-generation errors

Outline

Proposed method

- Weighting candidates vs. weighting component words
  - **Words** are easier to extract, match and weight
  - Useful for reducing over-generation errors
- Ensure that the importance of each word is counted only once in the set of keyphrases
  - Keyphrases should be extracted as a set rather than independently
- Finding the optimal set of keyphrases $\rightarrow$ combinatorial optimisation problem
  - Formulated as an integer linear problem (ILP)
  - Solved exactly using off-the-shelf solvers

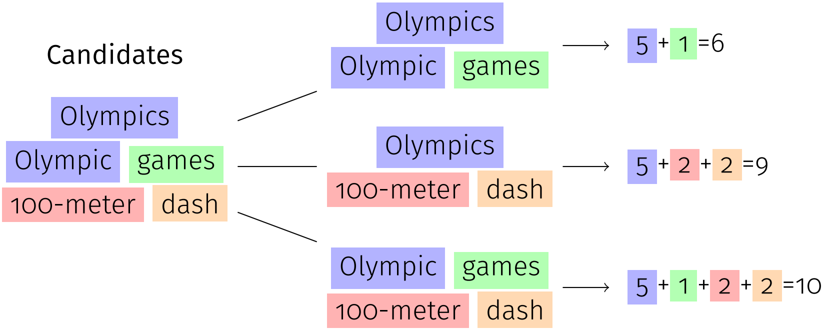
ILP model definition

- Based on the concept-based model for summarization [Gillick and Favre, 2009]
  - The value of a set of keyphrases is the sum of the weights of its unique words

Word weights

olympic(s) = 5
game = 1
100-meter = 2
dash = 2

Candidates



ILP model definition (cont.)

- Let x_i and c_j be binary variables indicating the presence of word i and candidate j in the set of extracted keyphrases

$$
\begin{aligned}
\max \quad & \sum_i w_i x_i && \leftarrow \text{Summing over unique word weights} \\
s.t. \quad & \sum_j c_j \leq N && \leftarrow \text{Number of extracted keyphrases} \\
& c_j Occ_{ij} \leq x_i, \quad \forall i, j && \leftarrow \text{Constraints for consistency} \\
& \sum_j c_j Occ_{ij} \geq x_i, \quad \forall i && Occ_{ij} = 1 \text{ if word } i \text{ is in candidate } j
\end{aligned}
$$

ILP model definition (cont.)

- By summing over word weights, the model overly favors long candidates
  - e.g. olympics < olympic games < modern olympic games
- To correct this bias in the model
  1. Pruning long candidates
  2. Adding constraints to prefer shorter candidates
  3. **Adding a regularization term to the objective function**

Regularization

- Let l_j be the size, in words, of candidate j, and $substr_j$ the number of times c_j occurs as a subtring in other candidates

$$\max \quad \sum_i w_i x_i - \lambda \sum_j \frac{(l_j - 1)c_j}{1 + substr_j}$$

- Regularization penalizes candidates made of more than one word, and is dampened for candidates that occur frequently as substrings

low λ ■■■; ■■■■; ■■■; ■■■■■; ■■■■

mid λ ■■; ■■■; ■■; ■■■; ■■

high λ ■; ■; ■; ■; ■

Outline

Introduction

Method

Experiments

Conclusion

Experimental parameters

- Experiments are carried out on the SemEval dataset [Kim et al., 2010]
  - Scientific articles from the ACM Digital Library
  - 144 articles (training) + 100 articles (test)
- Keyphrase candidates are sequences of nouns and adjectives
- Evaluation in terms of precision, recall and f-measure at the top N keyphrases
  - Sets of combined author- and reader-assigned keyphrases as reference keyphrases
  - Extracted/reference keyphrases are stemmed
- Regularization parameter λ tuned on the training set

Word weighting functions

- TF×IDF [Spärck Jones, 1972]
  - IDF weights are computed on the training set
- TextRank [Mihalcea and Tarau, 2004]
  - Window is sentence, edge weights are co-occurrences
- Logistic regression [Hong and Nenkova, 2014]
  - Reference keyphrases in training data are used to generate positive/negative examples
  - Features: position first occurrence, TF×IDF, presence in first sentence

Baselines

- `sum` : ranking candidates using the sum of the weights of their component words [Wan and Xiao, 2008]
- `norm` : ranking candidates using the sum of the weights of their component words normalized by their lengths
- Redundant keyphrases are pruned from the ranked lists
  1. Olympic games
  2. ~~Olympics~~
  3. 100-meter dash
  4. $\cdots$

Results

Weighting + Ranking	Top-5 candidates P	R	F	Top-10 candidates P	R	F
TF×IDF + `sum`	5.6	1.9	2.8	5.3	3.5	4.2
+ `norm`	19.2	6.7	9.9	15.1	10.6	12.3
+ `ilp`	25.4	9.1	13.3†	17.5	12.4	14.4†
TextRank + `sum`	4.5	1.6	2.3	4.0	2.8	3.3
+ `norm`	18.8	6.6	9.6	14.5	10.1	11.8
+ `ilp`	22.6	8.0	11.7†	17.4	12.2	14.2†
Logistic regression + `sum`	4.2	1.5	2.2	4.7	3.4	3.9
+ `norm`	23.8	8.3	12.2	18.9	13.3	15.5
+ `ilp`	29.4	10.4	15.3†	19.8	14.1	16.3

Results (cont.)

Method	Top-5 candidates				Top-10 candidates			
	P	R	F	rank	P	R	F	rank
SemEval - TF×IDF	22.0	7.5	11.2		17.7	12.1	14.4	
TF×IDF + `ilp`	25.4	9.1	13.3	14/20	17.5	12.4	14.4	18/20
SemEval - MaxEnt	21.4	7.3	10.9		17.3	11.8	14.0	
Logistic regression + `ilp`	29.4	10.4	15.3	10/20	19.8	14.1	16.3	15/20

Example (J-3.txt)

TF×IDF + **`sum`** (P $= 0.1$)
advertis bid; certain advertis budget; keyword bid; convex hull landscap; budget optim bid; **uniform bid strategi**; advertis slot; advertis campaign; ward advertis; searchbas advertis
TF×IDF + **`norm`** (P $= 0.2$)
advertis; advertis bid; **keyword**; keyword bid; landscap; advertis slot; advertis campaign; ward advertis; searchbas advertis; advertis random
TF×IDF + **`ilp`** (P $= 0.4$)
click; **advertis**; uniform bid; landscap; **auction**; convex hull; **keyword**; **budget optim**; single-bid strategi; queri

Outline

Conclusion

- Proposed ILP model
  - Can be applied on top of any word weighting function
  - Reduces over-generation errors by weighting candidates as a set
  - Substancial improvement over commonly used word-based ranking approaches
- Future work
  - Phrase-based model regularized by word redundancy

Thank you

florian.boudin@univ-nantes.fr

References I

Gillick, D. and Favre, B. (2009).
A scalable global model for summarization.
In Proceedings of the Workshop on Integer Linear Programming for Natural Language Processing, pages 10–18, Boulder, Colorado. Association for Computational Linguistics.

Hasan, K. S. and Ng, V. (2014).
Automatic keyphrase extraction: A survey of the state of the art.
In Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1262–1273, Baltimore, Maryland. Association for Computational Linguistics.

References II

Hong, K. and Nenkova, A. (2014).
Improving the estimation of word importance for news multi-document summarization.
In Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, pages 712–721, Gothenburg, Sweden. Association for Computational Linguistics.

Kim, S. N., Medelyan, O., Kan, M.-Y., and Baldwin, T. (2010).
Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles.
In Proceedings of the 5th International Workshop on Semantic Evaluation, pages 21–26, Uppsala, Sweden. Association for Computational Linguistics.

Mihalcea, R. and Tarau, P. (2004).
Textrank: Bringing order into texts.
In Lin, D. and Wu, D., editors, Proceedings of EMNLP 2004, pages 404–411, Barcelona, Spain. Association for Computational Linguistics.

References III

Spärck Jones, K. (1972).
A statistical interpretation of term specificity and its application in retrieval.
Journal of Documentation, 28:11–21.

Wan, X. and Xiao, J. (2008).
Collabrank: Towards a collaborative approach to single-document keyphrase extraction.
In Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008), pages 969–976, Manchester, UK. Coling 2008 Organizing Committee.