NRC·CNRC

NRC.CANADA.CA

LREC-COLING 2024

Keyphrase Generation: Lessons from a Reproducibility Study

Lingotto Conference Centre - Torino (Italia) 20-25 May, 2024

Edwin Thomas, Sowmya Vajjala

Task Overview

- Keyphrase Prediction (Extraction/Generation)
- Applications: Information Retrieval, Document Tagging, Text Summarization etc.
- Key-phrase Generation → Generate both PKP and AKP

Text: A *Framework to Automate the Parsing of Arabic Language Sentences This paper proposes a framework to automate the parsing (sic) of Arabic language sentences in general, although it focuses on the simple verbal sentences but it can be extended to any Arabic language sentence. The proposed system is divided into two separated phases which are lexical analysis and syntax analysis. Lexical phase analyses the words, finds its originals and roots, separates it from prefixes and suffixes, and assigns the filtered words to special tokens. Syntax analysis receives all the tokens and finds the best grammar for the given sequence of the tokens by using context free grammar. Our system assumes that the entered sentences are correct lexically and grammatically.*

Extractive: *lexical analysis, syntax analysis*

Abstractive: *Arabic language parser**

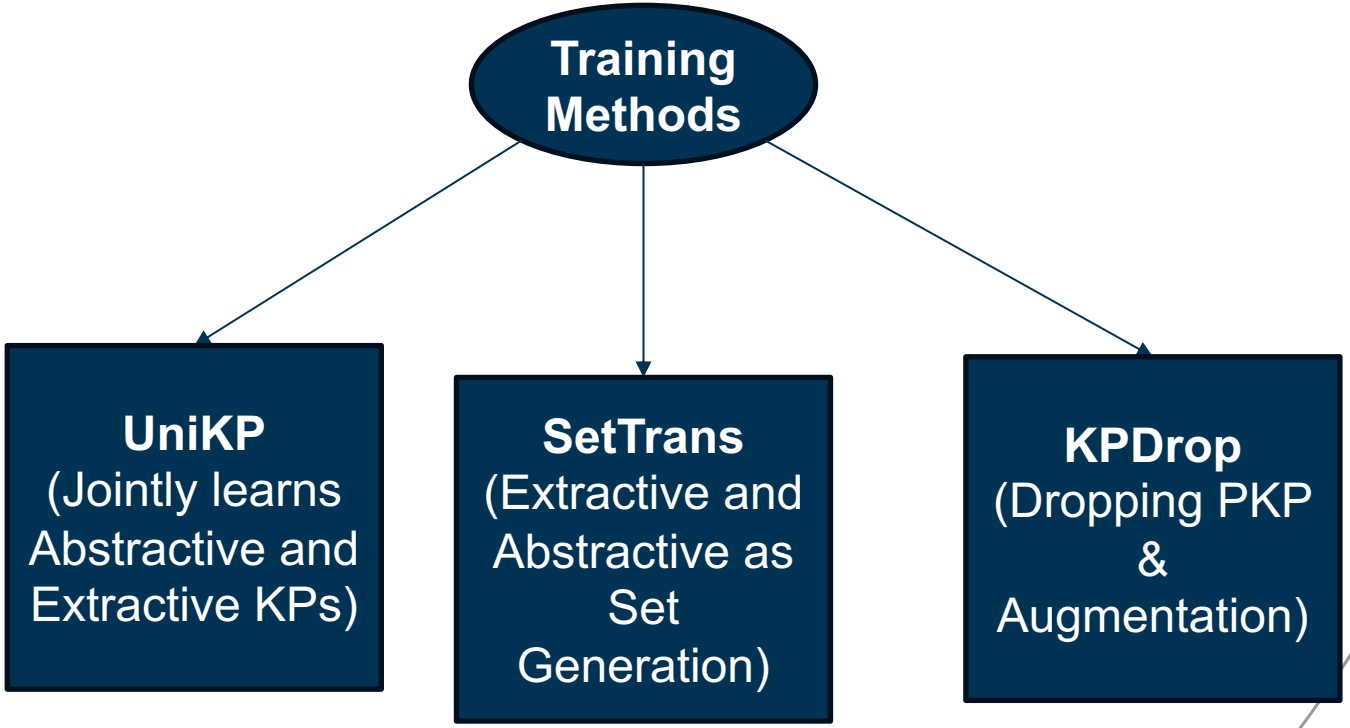
[does not appear in the text directly]*

Figure 1: An example from KP20K dataset (Meng et al., 2017) illustrating the task

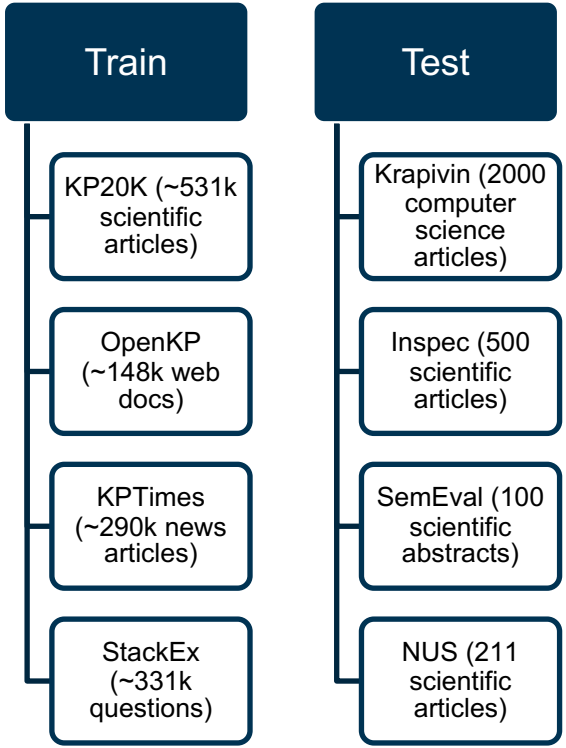
Relevance

- Despite **ample research** and **open datasets** in KPG there is minimal focus on **reproducibility**.
- We study 3 SOTA KPG approaches
- Questions we address:
  - To what extent can we reproduce existing results using the same code/model/datasets?
  - How does model performance vary when we use the same code/models, but train on different datasets instead?
  - How can we compare between two systems going beyond a single evaluation measure?

Methodology



Datasets & Evaluation Measures



- 4 Train sets
- 8 Test sets (including 4 train-test)

- Macro/Micro F1@K, F1@M, F1@O
- Similar metrics for Precision and Recall.

Reproducibility

- Reproduce default train/inference routines – 3 approaches
- **Minor differences:** H/W & S/W versions, undocumented prog. choices etc.
- **Major differences:** Reduced training epochs, points towards gap in original work and shared repos.

Dataset	UniKP		SetTrans		KPDrop	
	Pres	Abs	Pres	Abs	Pres	Abs
KP20K	↓7.2	↓3.6	↓2.0	↓1.4	↓1.2	↓1
Krapivin	-	-	↑1	↓1.6	↓0.0	↓0.2
Inspec	↑10.2	↓0.7	↓0.2	↓0.8	↓0.5	↓1.1
SemEval	↓4.9	↓2.5	↓4	↓0.6	↓3.4	↓0.6
NUS	↓7.7	↓2.4	↓1.5	↓2.3	↓1.2	↓1.9

Table 2: Difference in performance (in percentage) while reproducing (Macro-F1@*M* for KP-Drop/SetTrans; Micro-F1@*M* for UniKP)

Comparison of Decoding Strategies

- KPG research primarily uses 2 decoding strategies (**Greedy** & **Beam Search**).
- Beam Search variants: beam size = 5 and 50.
- *UniKP* → near-identical results (for both settings).
- *TransSet* and *KPDrop*
  - **Higher Beam Size** → Higher Recall & lesser overall performance.
  - **Lower Beam Size** → Lower Recall but better overall performance.
  - Beam Search has higher latencies than Greedy methods.

Greedy and Beam (n=5) offer comparable performances.

Reproducibility: other datasets

- Study performance of 3 KPG approaches on other datasets/domains.
- Train on OpenKP, KPTimes and StackEx datasets.
- **KPDrop** attains the **best performances**.

Only some *cross-domain transfer* exists when test datasets have intersecting domains with the train.

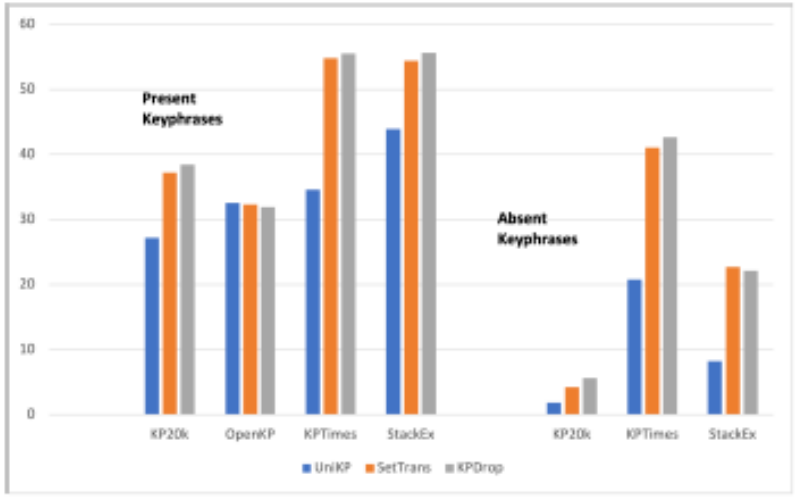


Figure 2: Performance of the three KPG models with other training datasets (Macro-F1@M)

Evaluation Measures

Micro vs Macro Average

- Studied differences between avg. macro and micro scores.
- No observable trend.

- **Recommendation**: Report both or Macro by convention.

R@10 vs 50

- Papers report R@50 → Retrieval focused applications.
- Higher Recall → irrelevant KPs

- **Recommendation:** Report both F-score and additional discussion on R@K.

F1@M vs O vs K

- F1@O is intuitively stricter.
- F1@O ~ F1@M (for low perf.)
- F1@M close to F1@K scores in most cases.
- F1 @O consistently lower → model over generates KPs.

- **Recommendation:** Report both F1@M and F1@O.

Significance Testing

- t-test are not appropriate for Precision and F-score as normality cannot be assumed **(Yeh, 2000).**

- Based on **Dror et al. (2018)** we perform *bootstrap and permutation tests*.

We recommend the bootstrap & permutation tests as they are easily available and don't have normality assumptions.

	F@5	F@M	F@O
Present			
System A	55.34**	55.49**	49.92
System B	54.63	54.77	49.34
Absent			
System A	42.36**	42.64**	37.68**
System B	40.58	41.02	35.58

Table 7: Significance Testing across evaluation measures (** indicates that the difference between System A and B is significant ($p < 0.001$)

Generation Overlap Analysis

- **Trend1:** When lesser overlap → more overlaps among KPs **not in GT.**
- **Trend2:** When higher overlap → more overlaps among KPs **in GT.**
- PKP and AKP plots almost identical (except freq.)

Trend 1, could also indicate that the GT has less coverage and model outputs are not incorrect.

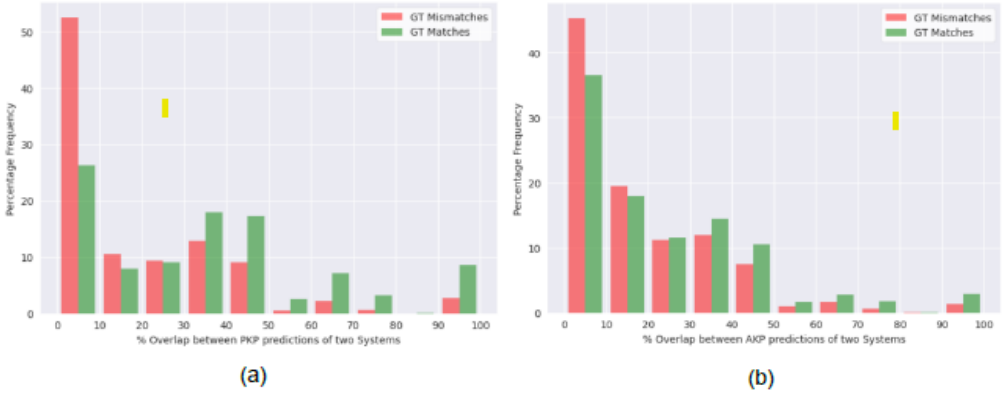


Figure 3: Percentage Overlap between keyphrase predictions generated by UniKP and KPDrop on KPTimes test set

Conclusions & Recommendations

Reproducing results even when code and data remain same is challenging.

When trained with same code but different training data, we observe large differences in performance trends.

The choice of reported evaluation measures impacts conclusions.

Recommendations:

- Report same evaluation measures for AKP/PKP and additional metrics separately.
- Specify whether micro/macro metrics is used.
- Employ appropriate statistical significance tests.

THANK YOU

Edwin Thomas • ethom123@uottawa.ca

Sowmya Vajjala • Sowmya.Vajjala@nrc-cnrc.gc.ca