From Fundamentals to Recent Advances
A Tutorial on Keyphrasification

Part 2.2 Deep Learning Methods for Keyphrase Generation

Rui Meng, Debanjan Mahata, Florian Boudin

ECIR 2022

Outline of Part II

Part I - Neural Keyphrase Extraction (Debanjan)

Part II - Neural Keyphrase Generation (Rui)

Part III - Hands-on Practice with OpenNMT-kpg and DLKP

Not All Keyphrases Are Extractable

- A non-negligible proportion of keyphrases are not present
  - Absent keyphrase: doesn't appear as a substring of the source text
  - Annotators assign keyphrases by their relevance/importance, not presence

Dataset	#Train	#Valid	#Test	Mean	Var	%Pre
KP20K	≈514k	≈20k	≈20k	5.3	14.2	63.3%
INSPEC	–	1500	500	9.6	22.4	78.5%
KRAPIVIN	–	1844	460	5.2	6.6	56.2%
NUS	–	-	211	11.5	64.6	51.3%
SEMEVAL	–	144	100	15.7	15.1	44.5%
STACKEX	≈298k	≈16k	≈16k	2.7	1.4	57.5%

(Yuan et al. 2018). One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases. ACL.

Not All Keyphrases Are Extractable

- Absent keyphrase
  - w.r.t lexical overlap between the source text and absent phrases
    - Reordered: words appear in different orders (e.g. “Information Sharing” vs “share information”).
    - Mixed: some words can be found in text (e.g. “Information Retrieval”).
    - Unseen: all words do not occur in the source document (e.g. “Retrieval Support”).

Study on the Structure of Index Data for Metasearch System

This paper proposes a new technique for Metasearch system, which is based on the grouping of both keywords and URLs. This technique enables metasearch systems to share information and to reflect the estimation of users’ preference. With this system, users can search not only by their own keywords but by similarity of HTML documents. In this paper, we describe the principle of the grouping technique as well as the summary of the existing search systems.

Present kps: Metasearch – Search System

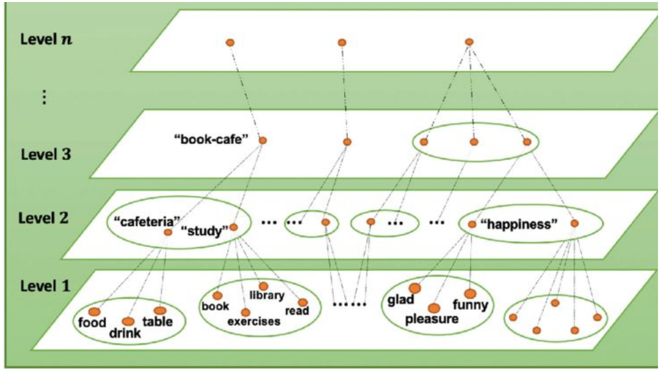
Absent kps: Information Sharing – Information Retrieval – User’s Behavior – Retrieval Support

Reordered Mixed Mixed Unseen

(Boudin, 2020) F. Boudin, Y. Gallina. Redefining Absent Keyphrases and their Effect on Retrieval Effectiveness. NAACL 2021.

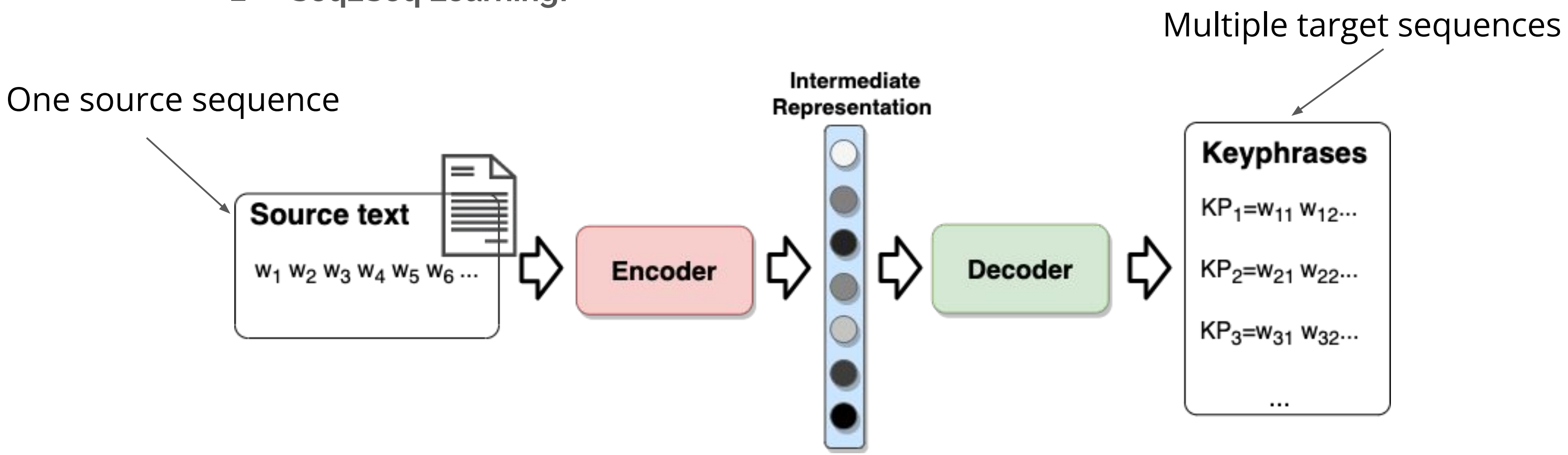
Not All Keyphrases Are Extractable

- Absent keyphrase
  - w.r.t functions
    - Higher-level concepts, i.e. biology, computer science, politics.
    - Generic in-domain terms, i.e. paper, design, model.
    - Synonyms/acronyms of present phrases, i.e. Ecommerce vs. electronic commerce
    - Others: beginner (in StackExchange, indicating a beginner question)

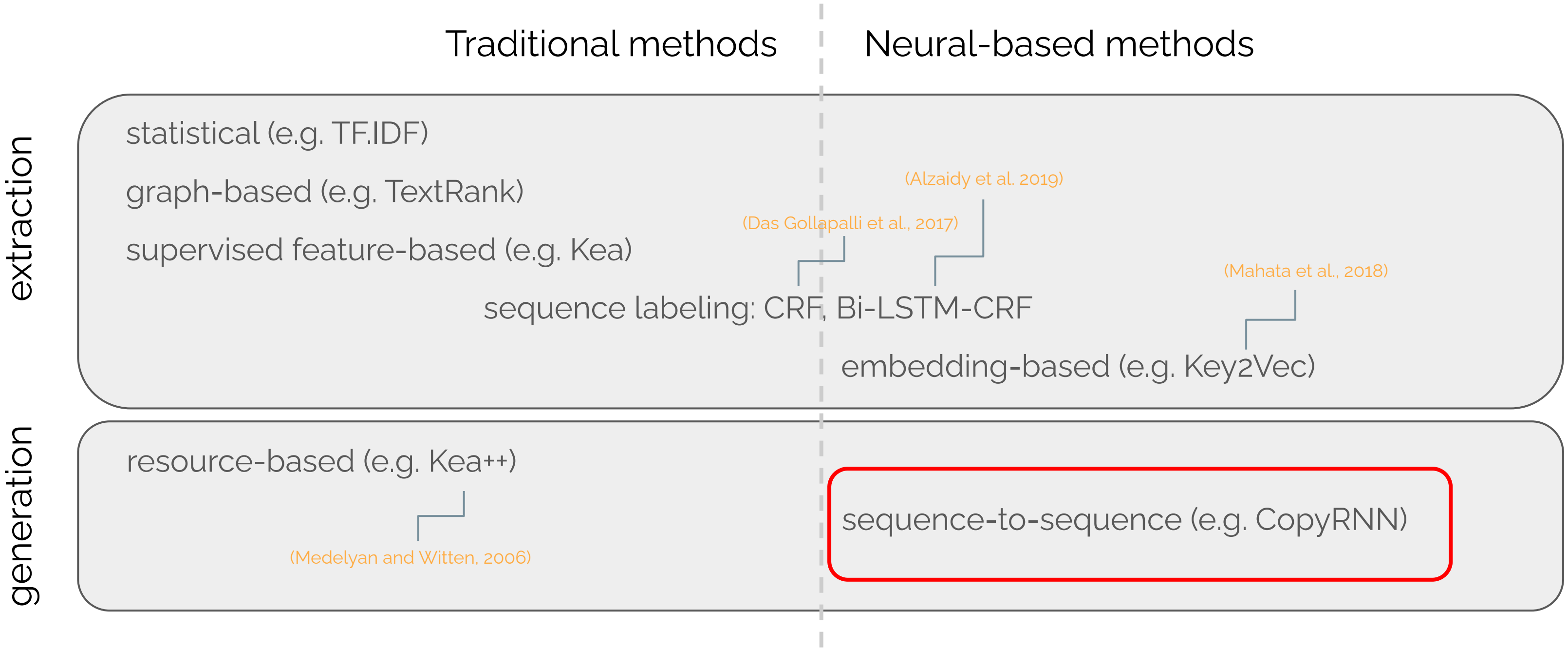


Neural Keyphrase Generation

- Predicting keyphrases as language generation
  - Each keyphrase is actually a short sequence of tokens
  - We can train neural networks to learn to generate phrases in a data-driven way
    - Input: a **SEQ**uence of source text
    - Output: multiple **SEQ**uences of tokens, each sequence is a keyphrase
    - **Seq2Seq Learning!**

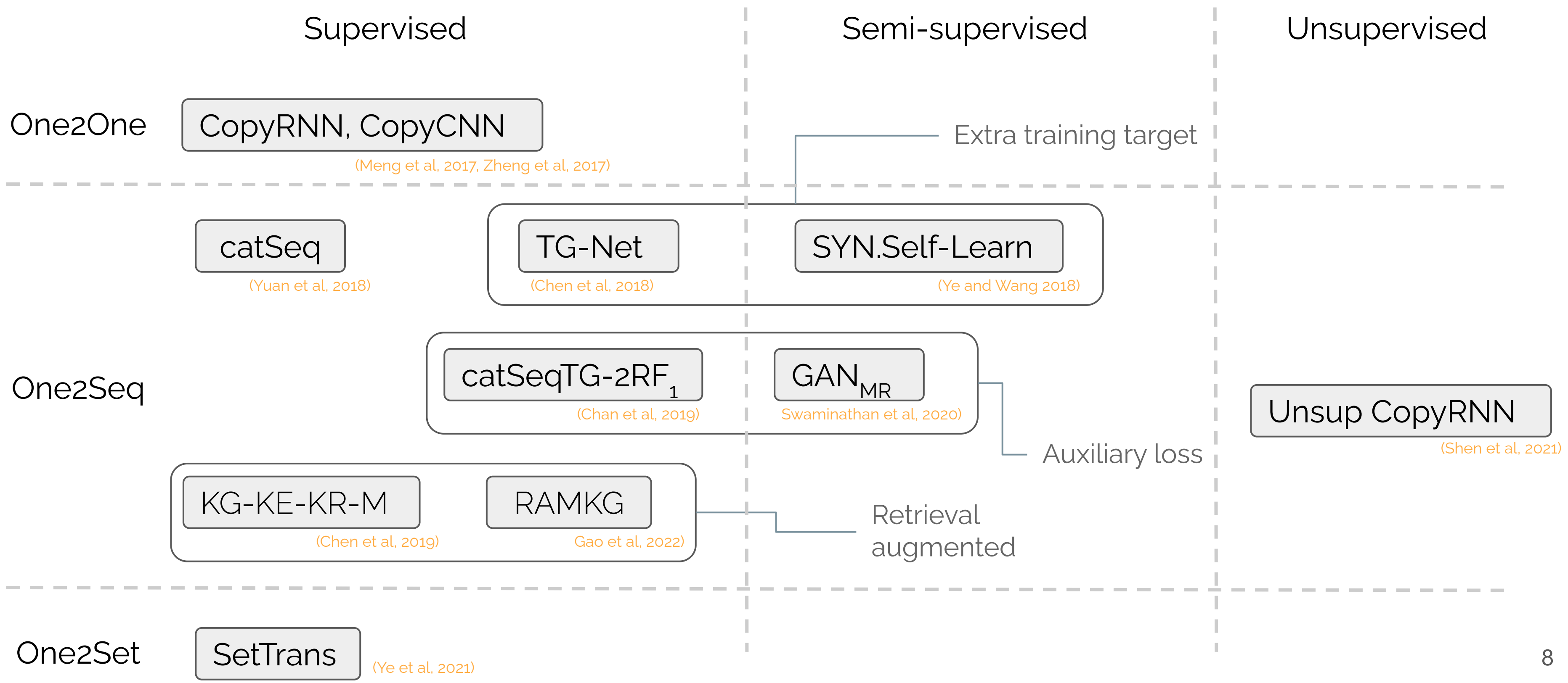


Taxonomy of Methods

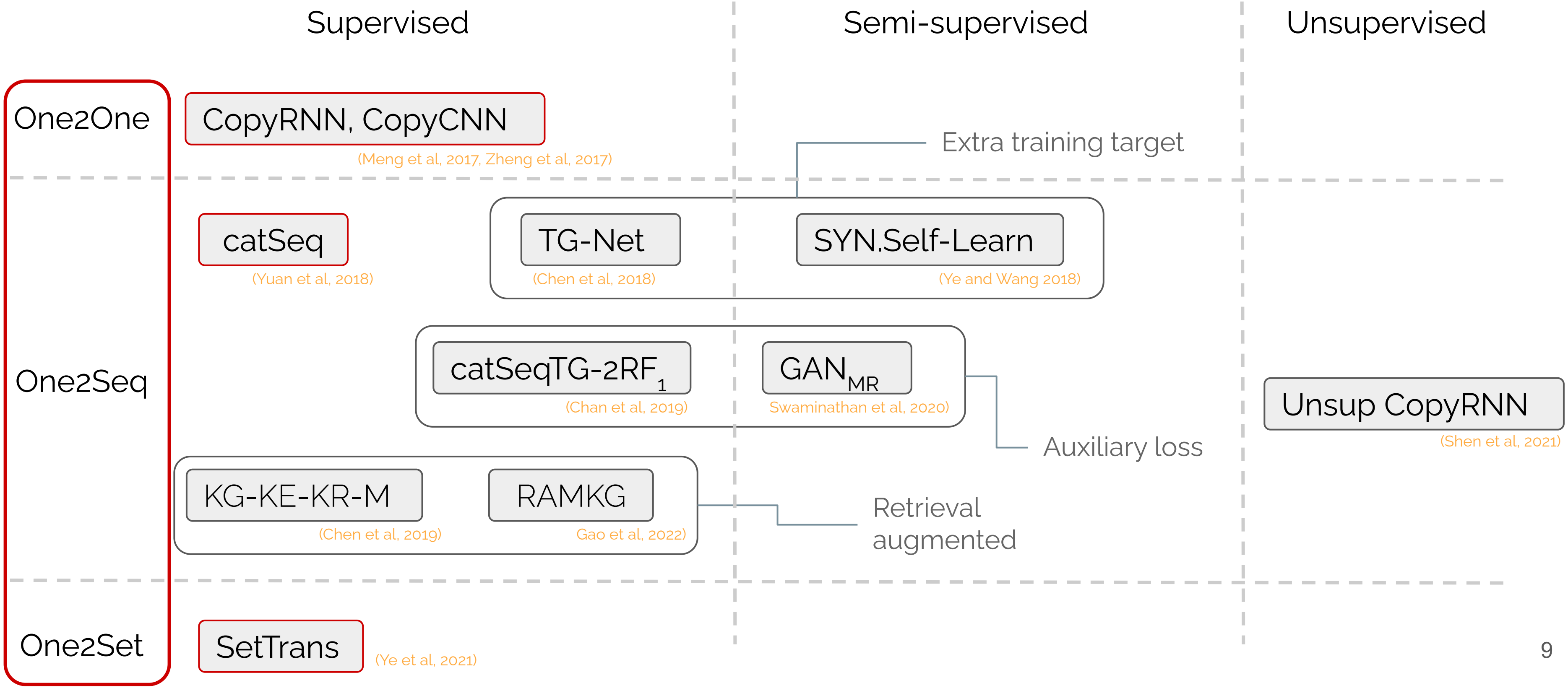


(Medelyan and Witten, 2006) Thesaurus based automatic keyphrase indexing. JCDL.
(Das Gollapalli et al., 2017) Incorporating expert knowledge into keyphrase extraction. AAAI.
(Mahata et al., 2018) Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings. NAACL.
(Alzaidy et al. 2019) Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. WWW.

Taxonomy of Generative Methods



Taxonomy of Generative Methods



Keyphrase Generation (KPG)

- Three types of training paradigms
  - **One2One**
    - Output one single phrase at a time
  - **One2Seq**
    - Output a sequence of multiple phrases at a time
  - **One2Set**
    - Output a set of multiple phrases at a time

(Meng et al. 2017). Deep Keyphrase Generation. ACL.
(Yuan et al. 2018). One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases. ACL.
(Ye and Wang, 2018). Semi-Supervised Learning for Neural Keyphrase Generation. EMNLP.
(Meng et al. 2021). An Empirical Study on Neural Keyphrase Generation. NAACL.
(Ye et al. 2021) "One2Set: Generating Diverse Keyphrases as a Set. ACL.

KPG-One2One

- Data preparation - each data example is split to multiple text-keyphrase pairs
  - Source text is duplicated K times
  - Each pair contains only one keyphrase
  - Great waste in training, e.g. in KP20k 510K->2.78M

Original Data Point (k target phrases)

[Source]
Language-specific Models in Multilingual Topic Tracking. Topic tracking is complicated when the stories in the stream occur in multiple languages. Typically, researchers have trained only English topic models because the training stories have been provided in English. In tracking, non-English test stories are then machine translated into English to compare them with the topic models. …

[Target]
[**classification**, crosslingual, Arabic, TDT, topic tracking, multilingual]

Src-Tgt Pair for Training (k pairs)

[Source] Language-specific Models in Multilingual Topic Tracking.…
[Target] <s> classification </s>

[Source] Language-specific Models in Multilingual Topic Tracking.…
[Target] <s> crosslingual </s>

[Source] Language-specific Models in Multilingual Topic Tracking.…
[Target] <s> arabic </s>

[Source] Language-specific Models in Multilingual Topic Tracking.…
[Target] <s> TDT </s>

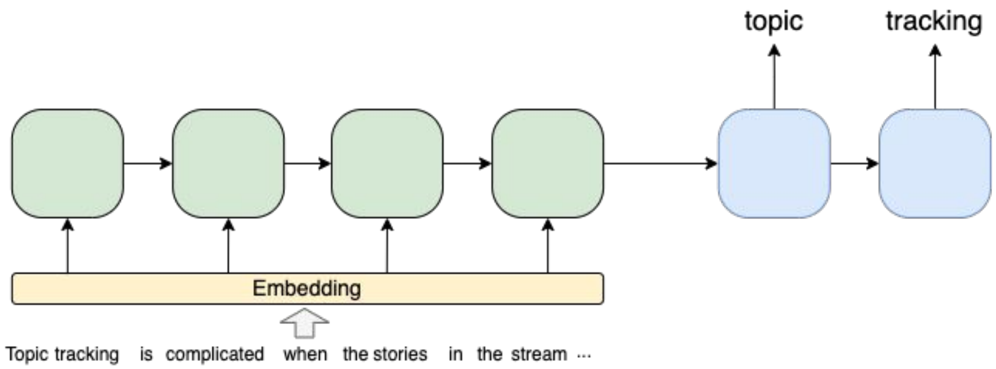
[Source] Language-specific Models in Multilingual Topic Tracking.…
[Target] <s> topic tracking </s>

[Source] Language-specific Models in Multilingual Topic Tracking.…
[Target] <s> multilingual </s>

(Meng et al. 2017). Deep Keyphrase Generation. ACL.

KPG-One2One

- Training
  - Model is trained to predict ONE target phrase at a time
    - Decoder only focuses on generating one phrase

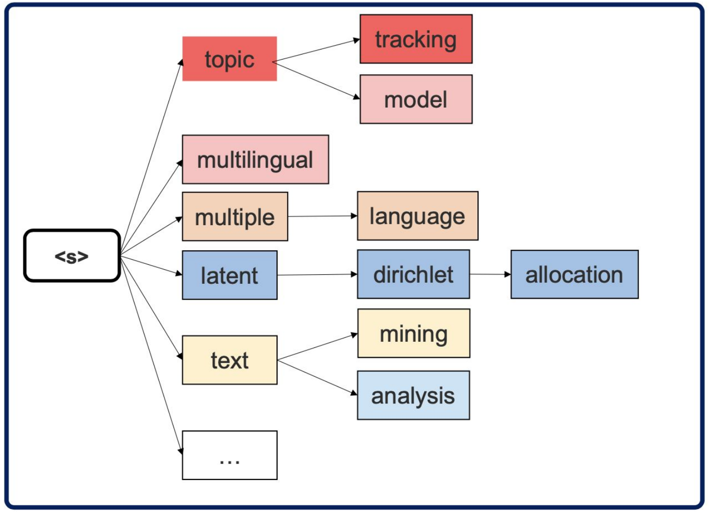


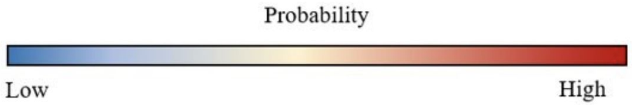
(Meng et al. 2017). Deep Keyphrase Generation. ACL.

KPG Inference

- How to generate multiple phrases?
  - Beam search is the key to produce a large number of unique phrases (up to 200)

Beam Search





KPG Inference

- Problems with One2One
  - Generated phrases are independent of each other and lack of diversity

Predicted Phrase	Score
1. jackson nets	[-1.5777]
2. jackson types	[-2.4076]
3. information system	[-3.7591]
4. business processes	[-4.0738]
5. jackson network	[-4.0782]
6. jackson net	[-4.1740]
7. petri nets	[-4.2971]
…	
11. jackson form	[-5.2189]
12. jackson model	[-5.2925]
13. jackson	[-5.5100]
14. jackson analysis	[-5.7585]

KPG-One2Seq

- Data Preparation
  - Concatenate multiple target phrases as a sequence
  - The order of concatenation can be effective in performance

[Source Sequence]
Language-specific Models in Multilingual Topic Tracking. Topic tracking is complicated when the stories in the stream occur in multiple languages. Typically, researchers have trained only English topic models because the training stories have been provided in English. In tracking, non-English test stories are then machine translated into English to compare them with the topic models. …

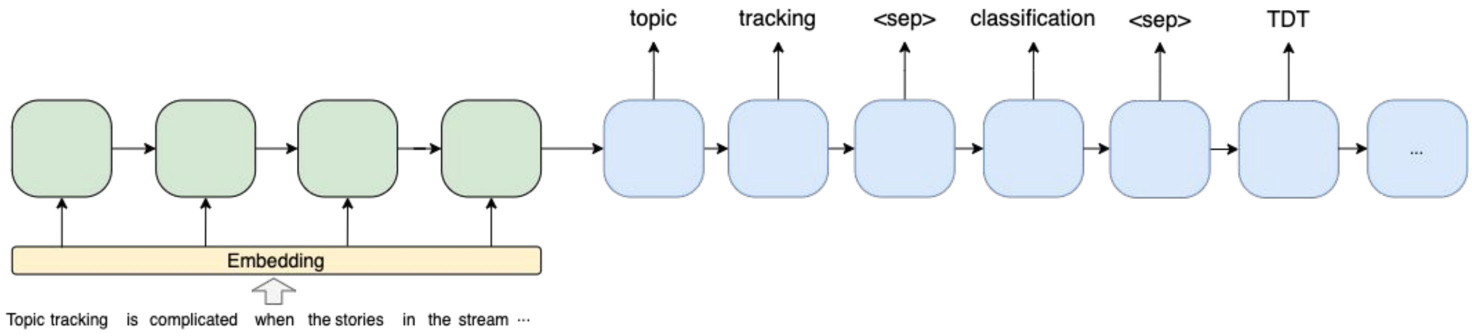
[Target Sequence]
[**classification**, crosslingual, Arabic, TDT, topic tracking, multilingual]

[Source] Language-specific Models in Multilingual Topic Tracking.…
[Target] <s> classification <sep> crosslingual <sep> Arabic <sep> TDT <sep> topic tracking <sep> multilingual </s>

(Yuan et al. 2018). One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases. ACL.
(Ye and Wang, 2018). Semi-Supervised Learning for Neural Keyphrase Generation. EMNLP.

KPG-One2Seq

- Training
  - Model is trained to predict a SEQuence of multiple phrases
    - More efficient and straightforward in training
    - Model can avoid generating similar phrases (w/ greedy decoding) since they are generated dependently



(Yuan et al. 2018). One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases. ACL.
(Ye and Wang, 2018). Semi-Supervised Learning for Neural Keyphrase Generation. EMNLP.

KPG Inference

- One2Seq can work with either greedy decoding or beam search
    - Greedy decoding
        - Special case of beam search (beam width=1)
        - Fast inference, but limited number of predicted phrases

topic tracking <sep> classification <sep> crosslingual <eos>

KPG Inference

- One2Seq can work with either greedy decoding or beam search
  - Beam search (beam width >> 1)
    - Aggregate predicted phrases from different sequences
    - Very inefficient due to many duplicates, e.g. width=50, 99% are duplicates

Decoding Results

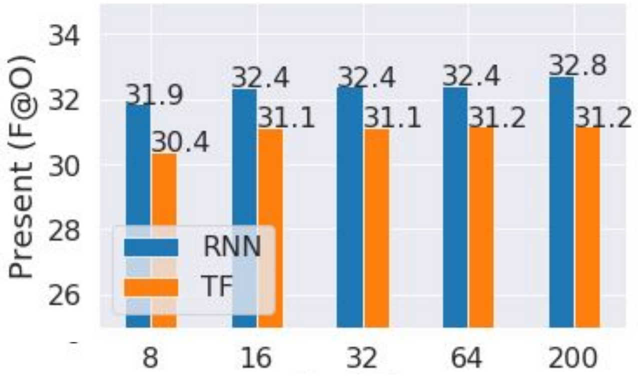
1. [-6.295] [**"fuzzy", "bayesian", "inference", "techniques"**, "<sep>", **"modus", "ponens", "rule"**, "<sep>", **"fuzzy", "sets"**]
2. [-6.295]["fuzzy", "bayesian", "inference", "<sep>", **"modus", "ponens", "rule"**, "<sep>", "fuzzy", "sets"]
3. [-6.295][**"decision", "making"**, "<sep>", "modus", "ponens", "rule", "<sep>", "fuzzy", "sets"]
4. [-6.743][**"fuzzy", "bayesian", "inference"**, "<sep>", "modus", "ponens", "rule", "<sep>", **"fuzzy", "inference"**]
5. [-6.822]["fuzzy", "bayesian", "inference", "techniques", "<sep>", "modus", "ponens", "rule", "<sep>", "fuzzy", "inference"],
6. [-7.128]["fuzzy", "bayesian", "inference", "techniques", "<sep>", "modus", "ponens", "rule", "<sep>", "fuzzy", "inference", "systems"],
7. [-7.421]["fuzzy", "bayesian", "inference", "techniques", "<sep>", "modus", "ponens", "rule", "<sep>", **"bayesian", "inference", "methods"**]
8. [-7.428]["fuzzy", "bayesian", "inference", "techniques", "<sep>", "decision", "making", "<sep>", "modus", "ponens", "rule", "<sep>", "fuzzy", "sets"]
9. ...

→

1. [-6.295] "fuzzy", "bayesian", "inference", "techniques"
2. [-6.295] "modus", "ponens", "rule"
3. [-6.295] "fuzzy", "sets"
4. [-6.295] "fuzzy", "bayesian", "inference"
5. [-6.295] "decision", "making"
6. [-6.743] "fuzzy", "inference"
7. [-7.128] "fuzzy", "inference", "systems"
8. [-7.421] "bayesian", "inference", "methods"
9. ...

KPG Results - Effects of Beam Width

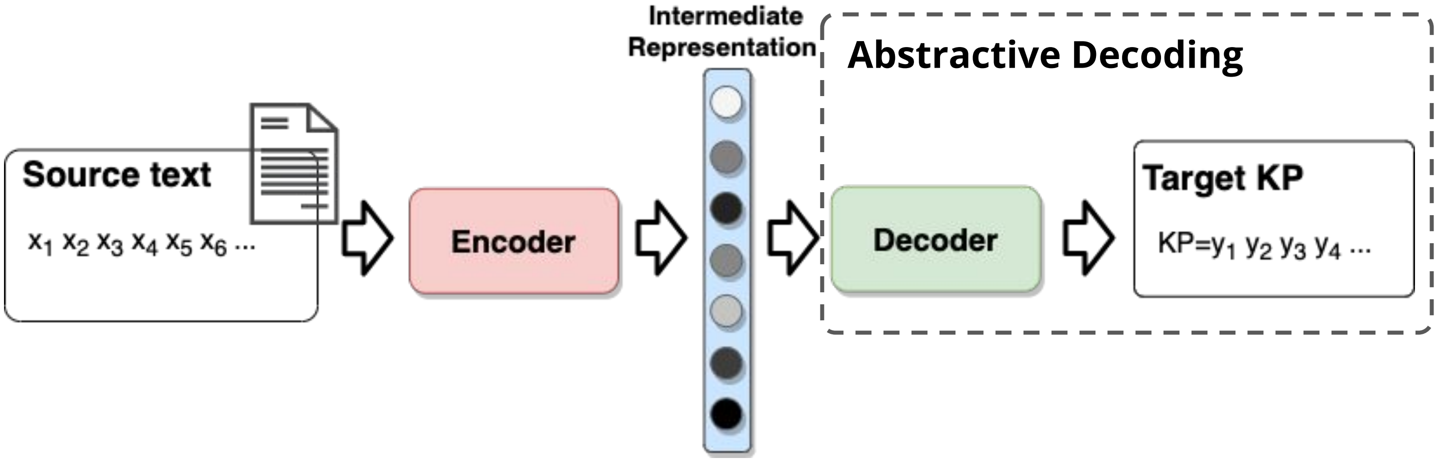
- Larger beam width is beneficial
  - especially for absent KPG
  - benefit gradually diminishes for present KPG
- Larger beam width -> greater computational cost, slower inference speed



(Meng et al. 2021). An Empirical Study on Neural Keyphrase Generation. NAACL.

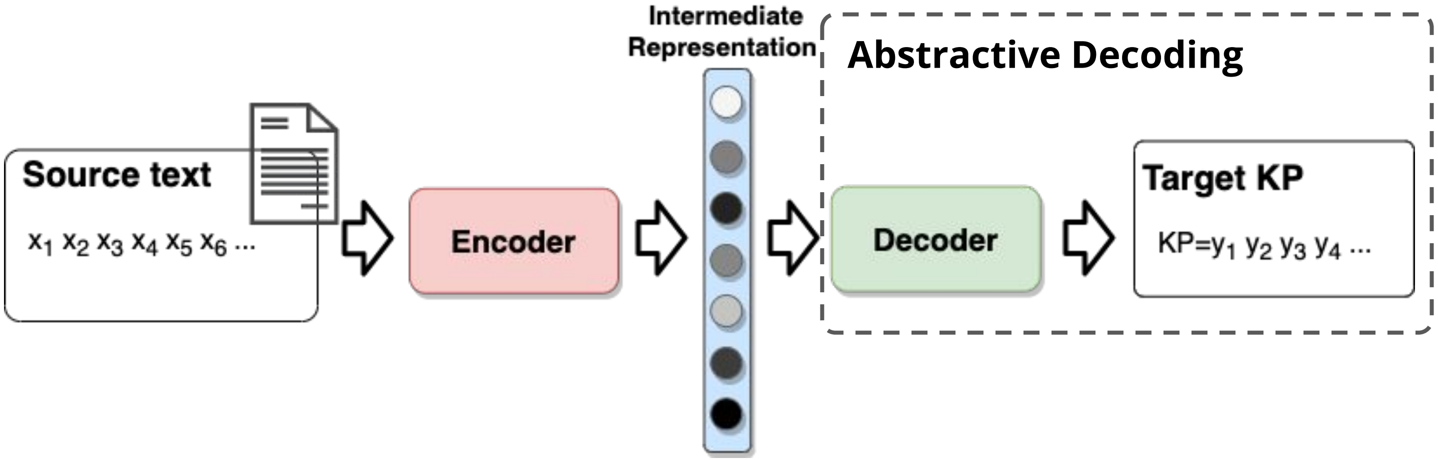
KPG Modeling

- Vanilla Seq2Seq
  - Generate target keyphrase abstractively



KPG Modeling

- Vanilla Seq2Seq
  - Generate target keyphrase abstractively

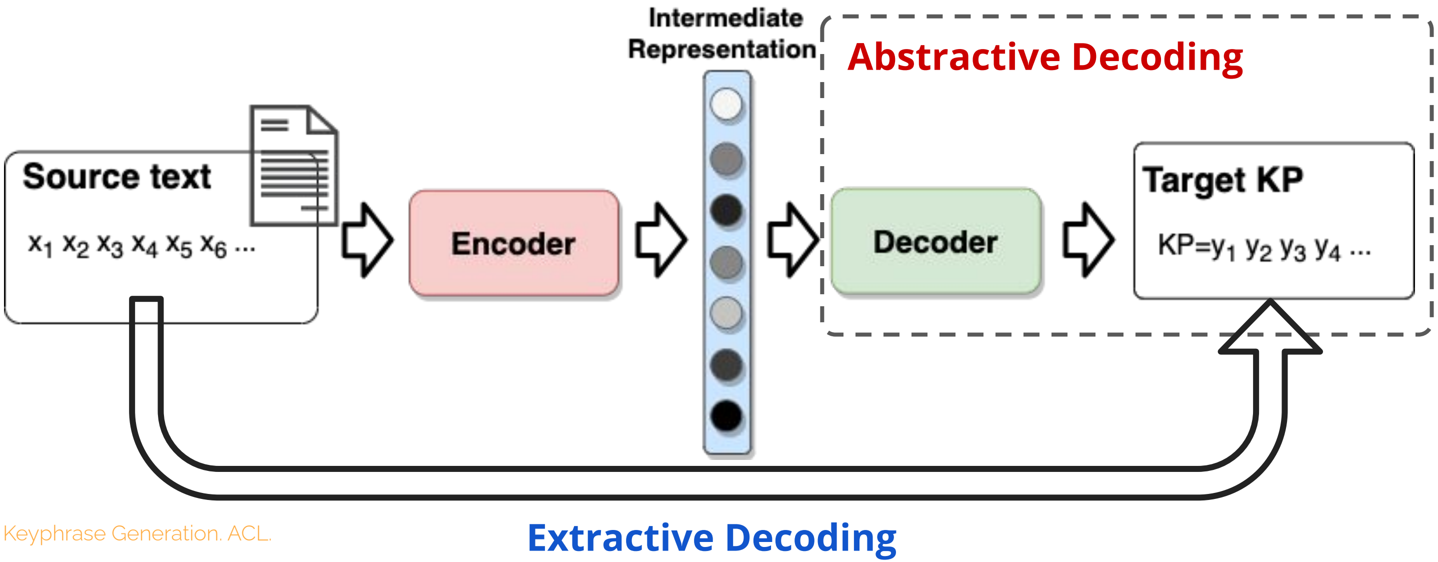


$$[x_i, x_{i+1}, \ldots, x_{i+k}] = [y_j, y_{j+1}, \ldots, y_{j+k}]$$

KPG Modeling

- Seq2Seq + Copy Attention
  - Generate target keyphrase both abstractively and extractively

$$P(w) = p_{abs} * \mathbf{P_{abs}}(w_{vocab}) + (1 - p_{abs}) * \mathbf{P_{ext}}(w_{src})$$

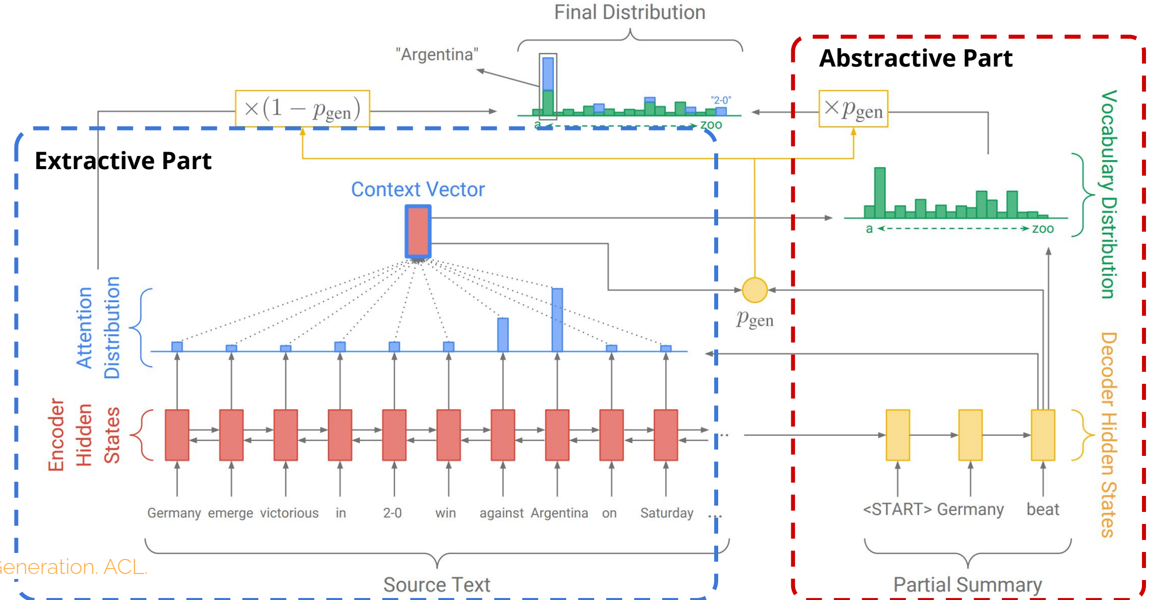


(Meng et al. 2017). Deep Keyphrase Generation. ACL.

KPG Modeling

- Copy Attention

$$P(w) = p_{abs} * \mathbf{P_{abs}}(w_{vocab}) + (1 - p_{abs}) * \mathbf{P_{ext}}(w_{src})$$

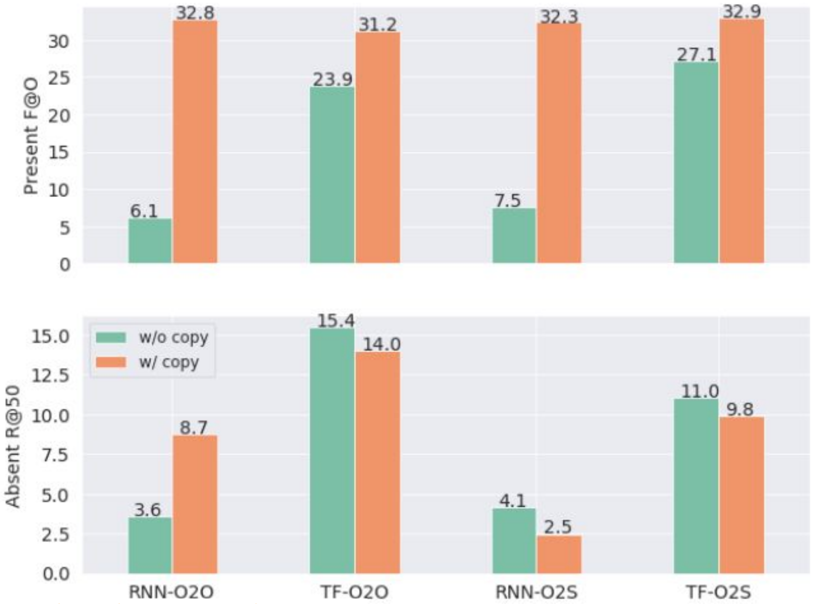


Picture credit to Abigail See
http://www.abigailsee.com/2017/04/16/taming-rnns-for-better-summarization.html

(Meng et al. 2017). Deep Keyphrase Generation. ACL.

KPG Results - Effects of Copy Attention

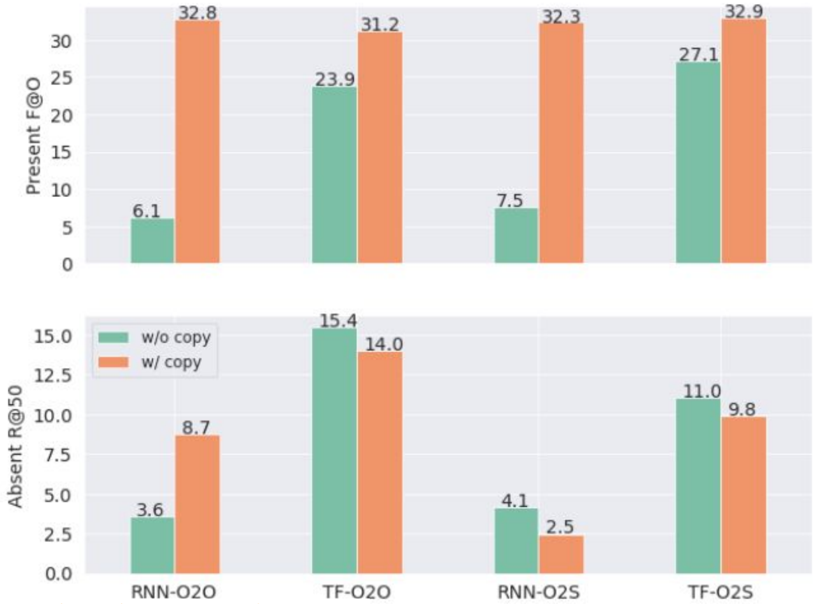
- Copy attention improves present performance significantly
  - Copy is necessary for RNN-based models
  - But can hurt Transformers on abstractiveness



(Meng et al. 2021). An Empirical Study on Neural Keyphrase Generation. NAACL.

KPG Results - Effects of Copy Attention

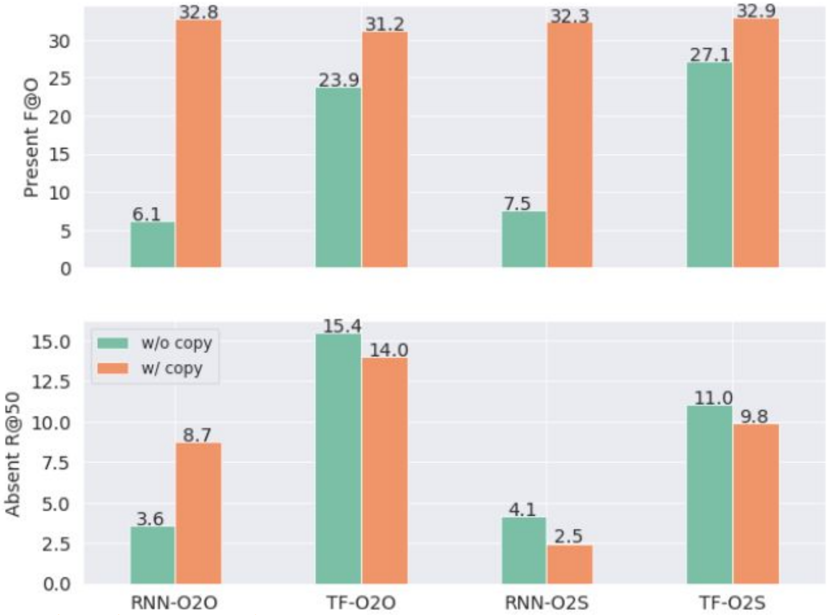
- One2One
  - Copy is necessary for RNN-based models
    - RNN+Copy outperforms Transformer
    - But can hurt Transformers on abstractiveness



(Meng et al. 2021). An Empirical Study on Neural Keyphrase Generation. NAACL.

KPG Results - Effects of Copy Attention

- One2Seq
  - One2Seq performs comparably to One2One on present phrases, but much poorer on absent
  - Copy shows smaller advantage with Transformer



(Meng et al. 2021). An Empirical Study on Neural Keyphrase Generation. NAACL.

KPG Results

- KPG outperforms classic extractive models by a large margin. Why?

Method	Inspec		Krapivin		NUS		SemEval		KP20k	
	F_1@5	F_1@10	F_1@5	F_1@10	F_1@5	F_1@10	F_1@5	F_1@10	F_1@5	F_1@10
Extractive Models										
Tf-Idf	22.3	30.4	11.3	14.3	13.9	18.1	12.0	18.4	10.5	13.0
TextRank	22.9	27.5	17.2	14.7	19.5	19.0	17.2	18.1	18.0	15.0
SingleRank	21.4	29.7	9.6	13.7	14.5	16.9	13.2	16.9	9.9	12.4
ExpandRank	21.1	29.5	9.6	13.6	13.7	16.2	13.5	16.3	N/A	N/A
Maui	4.0	3.3	24.3	20.8	24.9	26.1	4.5	3.9	26.5	22.7
KEA	10.9	12.9	9.6	13.6	6.8	8.1	2.7	2.7	18.0	16.3
Generative Models										
KPG	**28.5**	**32.5**	**32.0**	**27.0**	**40.2**	**35.9**	**32.9**	**34.6**	**33.1**	**27.9**
Gain%	**19.6%**	**6.9%**	**31.7%**	**29.8%**	**61.4%**	**37.5%**	**91.3%**	**88.0%**	**24.9%**	**22.9%**

(Meng et al. 2017). Deep Keyphrase Generation. ACL.

KPG Results

- KPG outperforms classic extractive models by a large margin. Why?
  - Better key-ness: phrases are ranked by model learned likelihood

Method	Inspec		Krapivin		NUS		SemEval		KP20k	
	F_1@5	F_1@10	F_1@5	F_1@10	F_1@5	F_1@10	F_1@5	F_1@10	F_1@5	F_1@10
Extractive Models										
Tf-Idf	22.3	30.4	11.3	14.3	13.9	18.1	12.0	18.4	10.5	13.0
TextRank	22.9	27.5	17.2	14.7	19.5	19.0	17.2	18.1	18.0	15.0
SingleRank	21.4	29.7	9.6	13.7	14.5	16.9	13.2	16.9	9.9	12.4
ExpandRank	21.1	29.5	9.6	13.6	13.7	16.2	13.5	16.3	N/A	N/A
Maui	4.0	3.3	24.3	20.8	24.9	26.1	4.5	3.9	26.5	22.7
KEA	10.9	12.9	9.6	13.6	6.8	8.1	2.7	2.7	18.0	16.3
Generative Models										
KPG	**28.5**	**32.5**	**32.0**	**27.0**	**40.2**	**35.9**	**32.9**	**34.6**	**33.1**	**27.9**
Gain%	19.6%	6.9%	31.7%	29.8%	61.4%	37.5%	91.3%	88.0%	24.9%	22.9%

(Meng et al. 2017). Deep Keyphrase Generation. ACL.

KPG Results

- KPG outperforms classic extractive models by a large margin. Why?
  - Better key-ness: phrases are ranked by model learned likelihood
  - Better phrase-ness: KPG learns how phrases are like from data, more reliable than N-grams/NPs

Tf-Idf

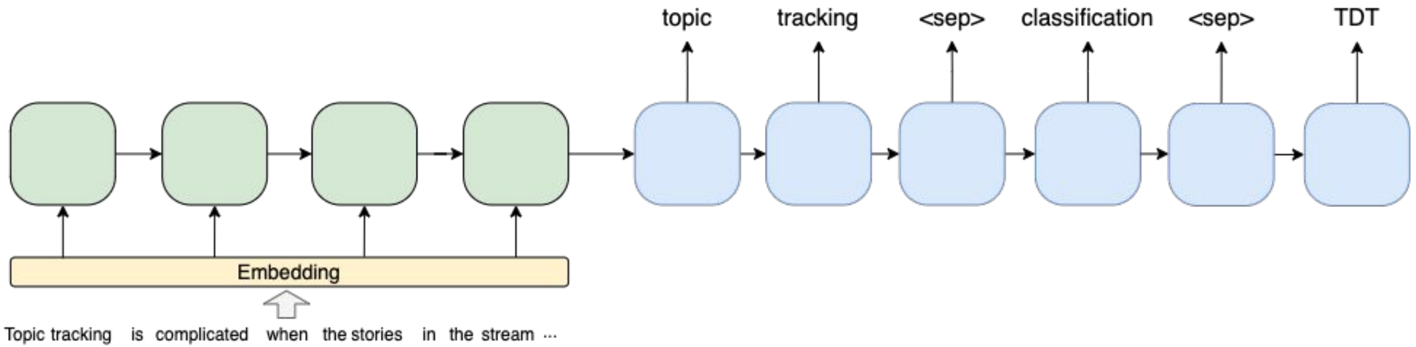
account
example
method
mixed central moment functions
moment function
nonlinear extrapolation
nonlinear extrapolation algorithm
nonlinear random dependences
problem
process
pugachev canonical decomposition apparatus
realization
s
scalar random process
third order

KPG

nonlinear extrapol
moment function
canon decomposit
extrapol algorithm
scalar random process
random process
central moment function
nonlinear extrapol algorithm
mix central moment function
central moment
mix central moment
random depend
investig process
nonlinear random depend
scalar random

KPG-One2Seq

- Pros
  - Training is straightforward and efficient
- Cons
  - The order for concatenating phrases can affect model performance
  - Poorer performance on absent phrases comparing with One2One



(Yuan et al. 2018). One Size Does Not Fit All: Generating and Evaluating Variable Number of Keyphrases. ACL.
(Ye and Wang, 2018). Semi-Supervised Learning for Neural Keyphrase Generation. EMNLP.

Case Study

[TITLE] A **Testability Measure** for **Hierarchical Design Environments**

[ABSTRACT] In this paper a new approach is proposed to compute testability of a **combinational circuit** in a hierarchical design environment. The testability of a circuit is first computed at the functional level using the **Walsh expression** of the functional block, and its complexity is linear with respect to the number of functional blocks. The functional level testability measure is then used to compute the testability at the gate/switch level. Our extensive simulation results show that the testability measure of the proposed method reflects closely to the actual testability measure (both at the **functional level** and the gate level) when the granularity of a functional block is much higher than that of primitive gates.

[GROUND-TRUTH]

Present KPs (#=8)

walsh expression
simulation
combinational circuit
testability measure
hierarchical design environments
gate switch level
functional level

Absent KPs (#=5)

walsh functions
logic testing
design for testability
logic cad
circuit cad

[PREDICT]

Present (Top 10)

1. hierarchical design
2. **testability measure** [correct!]
3. **combinational circuit** [correct!]
4. **walsh expression** [correct!]
5. testability
6. **hierarchical design environments** [correct!]
7. design environment
8. **functional level** [correct!]
9. functional block
10. design

Absent (Top 20)

1. logic design
2. design automation
3. **design for testability** [correct!]
4. **logic testing** [correct!]
5. **logic cad** [correct!]
6. circuit faults
7. automatic test pattern generation
8. vlsi
9. design principle
10. built in self test
11. circuit design
12. circuit synthesis
13. complexity theory
14. circuit simulation
15. **walsh functions** [correct!]
16. test measure
17. integrated circuit design
18. gate level testability
19. test
20. circuit analysis computing

Generating Absent KeyPhrases

[TITLE] How to predict on part of image after training on other part of image?

[QUESTION] I have images of identity cards (manually taken so not of same size) and I need to extract the text in it. I used tesseract to predict bounding boxes for each letter and am successful to some extent but some letters are not bounded.
So, I have around 5000 bounding boxes in all images combined. I want to train it so as to predict bounding boxes for remaining letters. After predicting the bounding boxes I will try to classify the image into characters. This is different from conventional machine learning problem where I donot have training and testing data separately.

[GROUND-TRUTH] (**#absent**=5)

neural network
deep learning
image classification
convnet
computer vision

[PREDICT] (Top 20)

1. [tesseract]
2. [machine learning]
3. **image classification** **[correct!]**
4. **deep learning** **[correct!]**
5. **computer vision** **[correct!]**
6. classification
7. tensorflow
8. untagged
9. image recognition
10. python
11. nlp
12. **neural network** **[correct!]**
13. **convnet** **[correct!]**
14. scikit learn
15. deep learning s
16. deep learningd
17. deep learning convnet
18. deep network
19. convnet s
20. deep learning s s

Case Study (Bad)

[TITLE] On the relationship between workflow models and document types

[ABSTRACT] The best practice in information system development is to model the business processes that have to be supported and the database of the information system separately. This is inefficient because they are closely related. Therefore we present a framework in which it is possible to derive one from the other. To this end we introduce a special class of **Petri nets**, called Jackson nets, to model the business processes, and a document type, called Jackson types, to model the database. We show that there is a one-to-one correspondence between Jackson nets and Jackson types. We illustrate the use of the framework by an example.

[GROUND-TRUTH]

Present KPs (#=1)

Petri net

Absent KPs (#=4)

Workflow management system
Document management system
Data type
Information system design methodology

[PREDICT]

Present

1. jackson nets
2. jackson types
3. information system
4. business processes
5. jackson net
6. **petri nets** [**correct**!]
7. model
8. workflow
9. database
10. jackson
11. relationship
12. information system development

Absent

1. jackson network
2. jackson form
3. jackson model
4. jackson analysis
5. jackson relation
6. jackson types networks
7. information system model
8. model driven development
9. jackson term
10. jackson network model
11. jackson network analysis
12. jackson type network
13. jackson term model
14. jackson types model

ECIR 2022

KPG-One2Seq: Order Matters

[Source Sequence]=title+abstract
Language-specific Models in **Multilingual** **Topic Tracking**. Topic tracking is complicated when the stories in the stream occur in multiple languages. Typically, researchers have trained only English topic models because the training stories have been provided in English. In tracking, non-English test stories are then machine translated into English to compare them with the topic models. ...

[Target Sequence]=keyphrases
[classification, crosslingual, Arabic, TDT, topic tracking, multilingual]

[Present Phrases] topic tracking, multilingual
[Absent Phrases] classification, crosslingual, Arabic, TDT

Random	**[Source]** Language-specific Models in Multilingual Topic **[Target]** <bos> TDT <sep> multilingual <sep> crosslingual <sep> Arabic <sep> classification <sep> topic tracking
Length	**[Source]** Language-specific Models in Multilingual Topic **[Target]** <bos> classification <sep> crosslingual <sep> Arabic <sep> TDT <sep> multilingual <sep> topic tracking
Original	**[Source]** Language-specific Models in Multilingual Topic **[Target]** <bos> classification <sep> crosslingual <sep> Arabic <sep> TDT <sep> topic tracking <sep> multilingual
Alpha	**[Source]** Language-specific Models in Multilingual Topic **[Target]** <bos> Arabic <sep>classification <sep> crosslingual <sep> multilingual <sep> TDT <sep> topic tracking
Abs-Pres	**[Source]** Language-specific Models in Multilingual Topic **[Target]** <bos> Arabic <sep> TDT <sep>classification <sep> crosslingual <sep> multilingual <sep> topic tracking
Pres-Abs	**[Source]** Language-specific Models in Multilingual Topic **[Target]** <bos> multilingual <sep> topic tracking <sep> TDT <sep> Arabic <sep> classification <sep> crosslingual

(Meng et al. 2021). An Empirical Study on Neural Keyphrase Generation. NAACL.

KPG-One2Seq: Order Matters

- Target phrase order shows distinct effects on performance (i.e. Pres-Abs >> Abs-Pres).
  - Column: models trained in different phrase concatenation order
  - Row: scores on six scientific paper datasets

(c) Greedy Decoding, Transformer

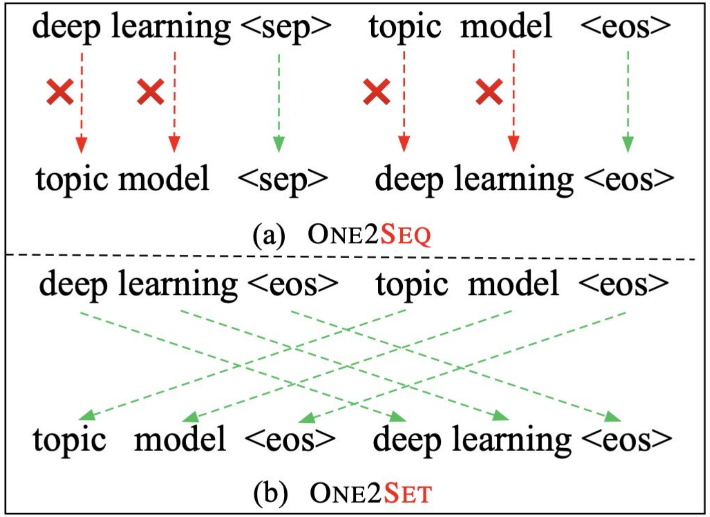
Diff between Pres-Abs and Abs-Pres:

- w/ RNN: 3.0
- w/ Transformer: 6.6

(Meng et al. 2021). An Empirical Study on Neural Keyphrase Generation. NAACL.

KPG-One2Set

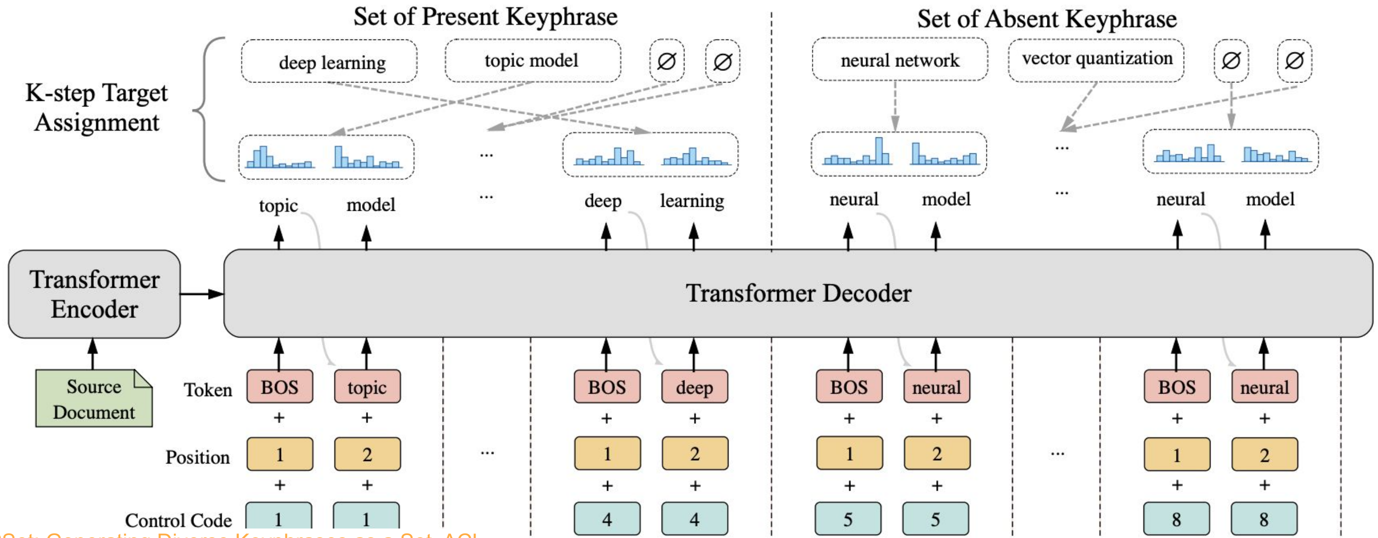
- Making phrase order less impactful
    - Utilize Non-autoregressive Decoding and Hungarian algorithm to eliminate the effect of phrase orders



(Ye et al. 2021) "One2Set: Generating Diverse Keyphrases as a Set. ACL.

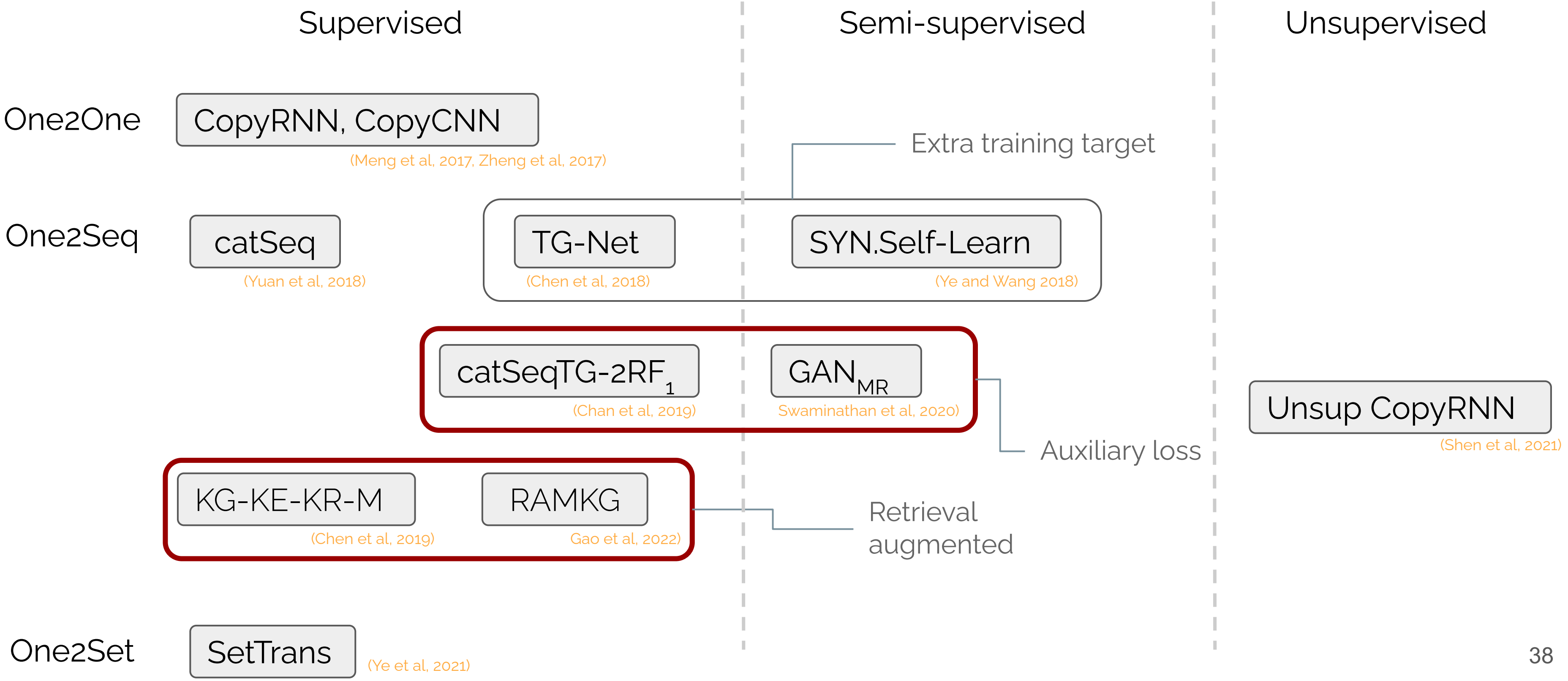
KPG-One2Set

- One2Set (Ye et al. 2021)
  - Utilize Non-autoregressive Decoding and Hungarian algorithm to eliminate the effect of phrase orders



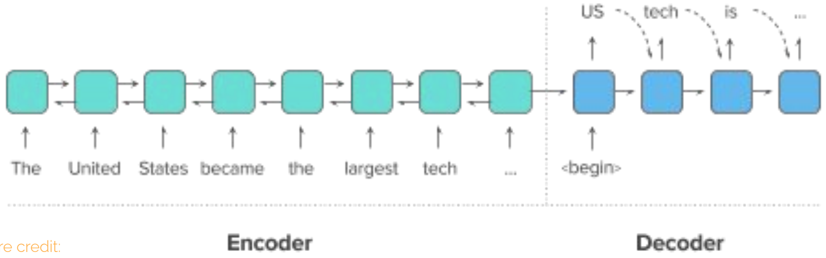
(Ye et al. 2021) "One2Set: Generating Diverse Keyphrases as a Set. ACL.

Taxonomy of Generative Methods



Learning to Generate Keyphrases Beyond MLE

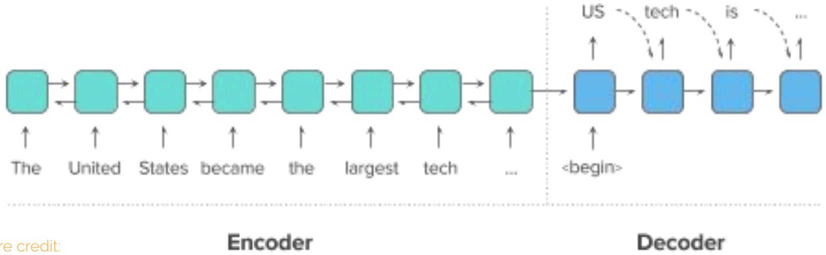
- Bridge the gap between MLE loss and keyphrase evaluation
  - Most keyphrase generation models are trained with MLE
    - Maximum-likelihood estimation: maximizing the probability of the next word
  - Keyphrases are evaluated as a set, such as F1-score

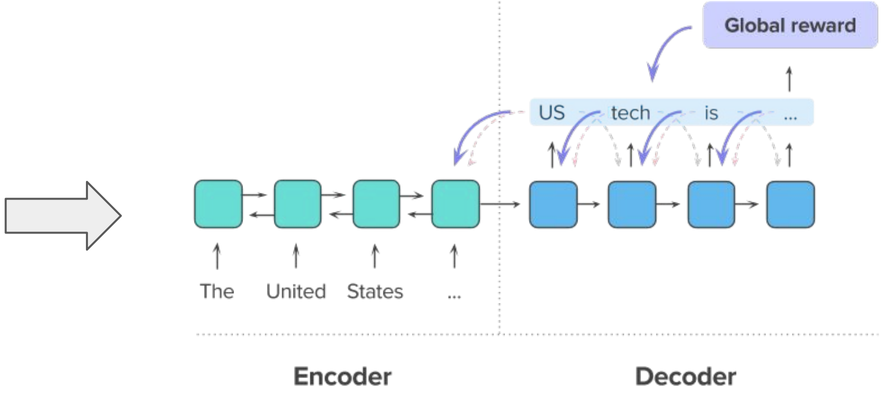


Picture credit: https://www.salesforce.com/products/einstein/ai-research/tl-dr-reinforced-model-abstractive-summarization/

Learning to Generate Keyphrases Beyond MLE

- Bridge the gap between MLE loss and keyphrase evaluation
  - Most keyphrase generation models are trained with MLE
    - Maximum-likelihood estimation: maximizing the probability of the next word
  - Keyphrases are evaluated as a set, such as F1-score
  - Utilizing Reinforcement Learning to infuse global reward to models





Picture credit: https://www.salesforce.com/products/einstein/ai-research/tl-dr-reinforced-model-abstractive-summarization/

Learning to Generate Keyphrases Beyond MLE

- Bridge the gap between MLE loss and keyphrase evaluation
  - Most keyphrase generation models are trained with MLE
    - Maximum-likelihood estimation: maximizing the probability of the next word
  - Keyphrases are evaluated as a set, such as F1-score
  - Utilizing Reinforcement Learning to infuse global reward to models
- Related work
  - Manually-designed reward
    - catSeqTG-2RF1, Chan et al. 2019
  - Learned reward via GAN
    - KPG-GAN$_{MR}$, Swaminathan et al. 2020

(Chan et al. 2019), Neural Keyphrase Generation via Reinforcement Learning with Adaptive Rewards." ACL.
(Swaminathan et al. 2020), A preliminary exploration of GANs for keyphrase generation. EMNLP

Keyphrase Generation using GANs

ACL Anthology

A Preliminary Exploration of GANs for Keyphrase Generation

Avinash Swaminathan, Haimin Zhang, Debanjan Mahata, Rakesh Gosangi, Rajiv Ratn Shah, Amanda Stent

Abstract

We introduce a new keyphrase generation approach using Generative Adversarial Networks (GANs). For a given document, the generator produces a sequence of keyphrases, and the discriminator distinguishes between human-curated and machine-generated keyphrases. We evaluated this approach on standard benchmark datasets. We observed that our model achieves state-of-the-art performance in the generation of abstractive keyphrases and is comparable to the best performing extractive techniques. Although we achieve promising results using GANs, they are not significantly better than the state-of-the-art generative models. To our knowledge, this is one of the first works that use GANs for keyphrase generation. We present a detailed analysis of our observations and expect that these findings would help other researchers to further study the use of GANs for the task of keyphrase generation.

PDF

Cite

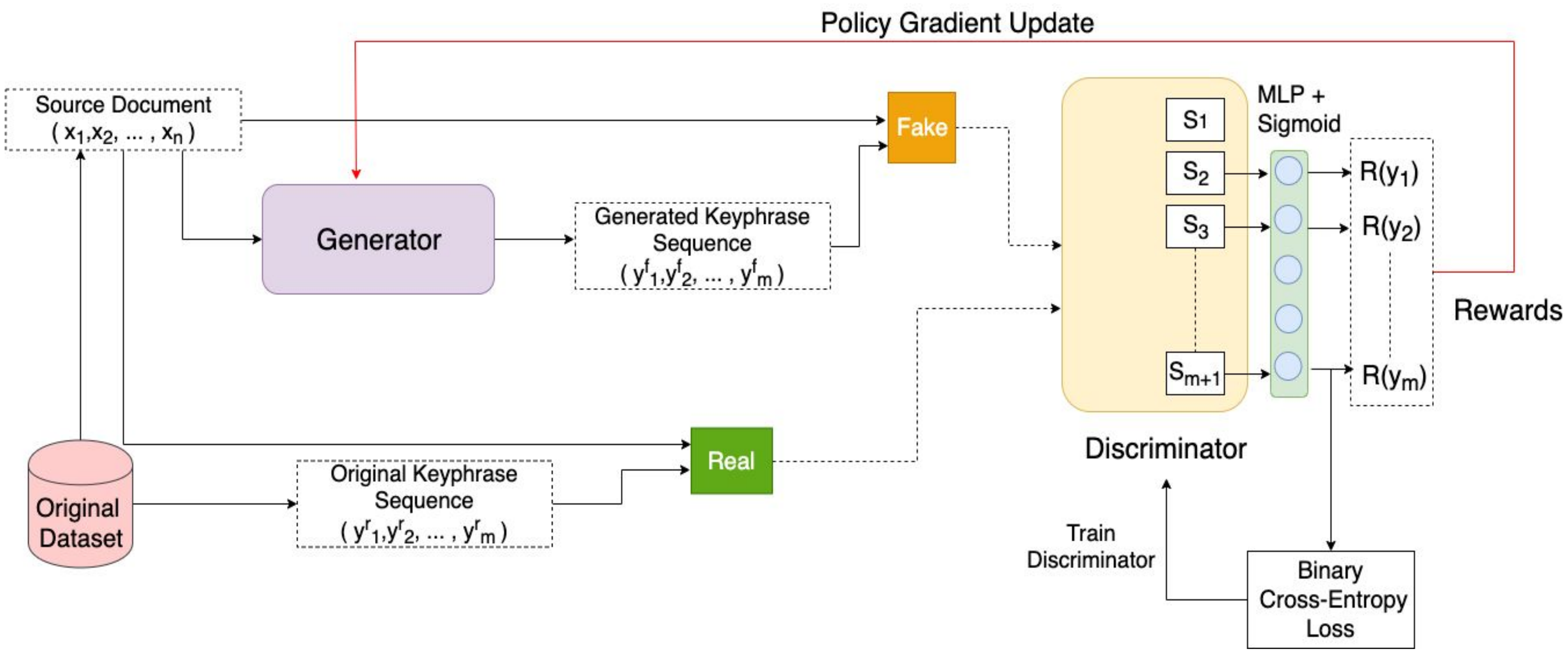
Search

Video

avinsit123 / **keyphrase-gan**

Swaminathan, A., Zhang, H., Mahata, D., Gosangi, R., Shah, R., & Stent, A. (2020, November). A preliminary exploration of GANs for keyphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 8021-8030).

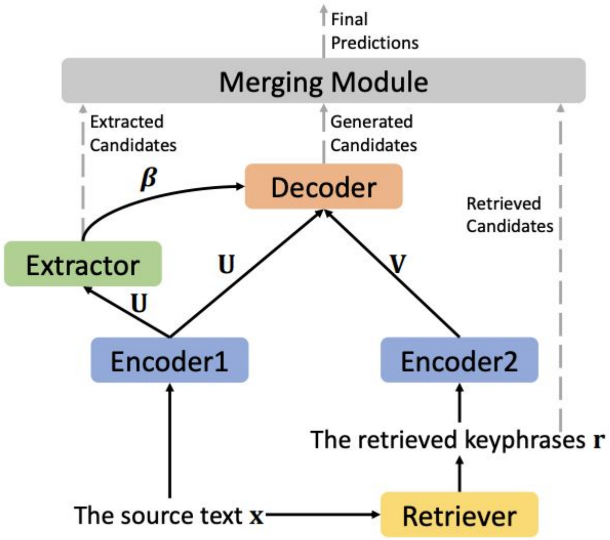
KPG via GAN



Swaminathan, A., Zhang, H., Mahata, D., Gosangi, R., Shah, R., & Stent, A. (2020, November). A preliminary exploration of GANs for keyphrase generation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)* (pp. 8021-8030).

Retrieval-Augmented Keyphrasification

- Retrieve relevant data as external knowledge
  - KG-KE-KR-M (Chen et al. 2019)
    - Retrieve similar documents and use their associated keyphrases as external knowledge for the generative model

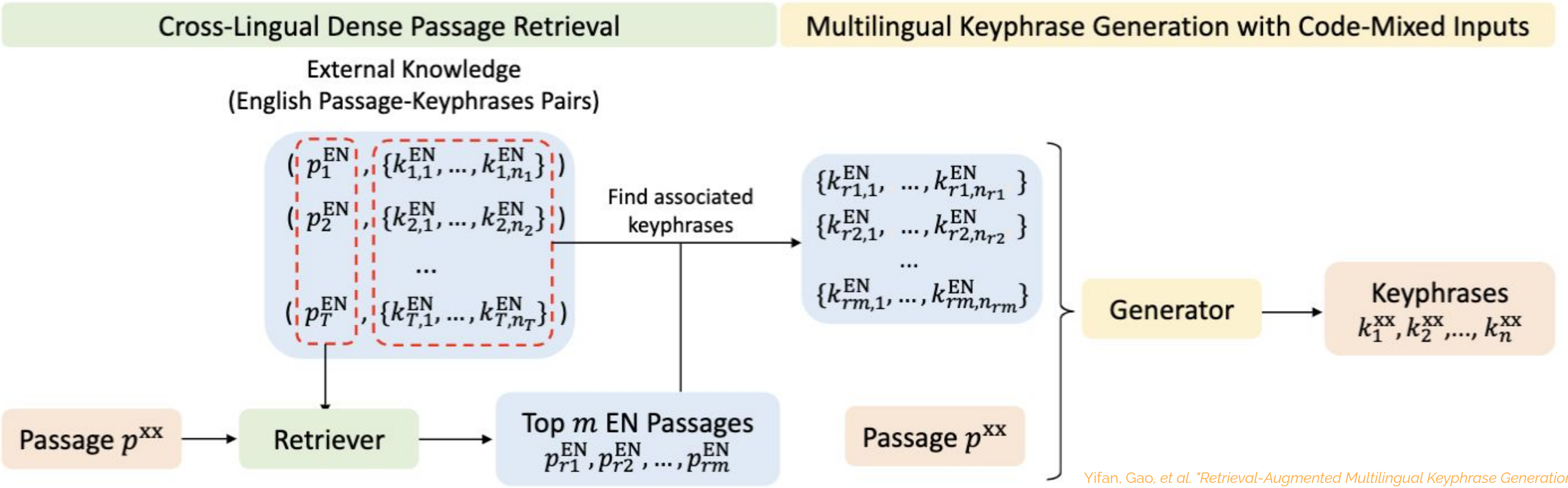


Candidate Sources	Total F_1@10	Present F_1@5	Absent R@10
gk, ek, rk	**0.250±0.002**	**0.330±0.002**	**0.172±0.002**
gk, ek	0.249±0.003	0.328±0.003	0.154±0.002
gk, rk	0.249±0.002	0.329±0.002	0.172±0.002
gk	0.248±0.003	0.327±0.003	0.154±0.002

(Chen, 2019) An Integrated Approach for Keyphrase Generation via Exploring the Power of Retrieval and Extraction NAACL.

Retrieval-Augmented Keyphrasification

- Retrieve relevant data as external knowledge
  - RAMKG for multilingual keyphrase generation (Gao et al. 2022)
    - Retrieve passage-phrase pairs in English via dense retriever (mBERT)
    - Alleviate the resource scarcity issue in low-resource languages



Yifan, Gao, *et al.* *"Retrieval-Augmented Multilingual Keyphrase Generation with Retriever-Generator Iterative Training."* NAACL findings. 2022.

Retrieval-Augmented Keyphrasification

- Retrieve relevant data as external knowledge
  - RAMKG for multilingual keyphrase generation (Gao et al. 2022)
    - Retrieve passage-phrase pairs in English via dense retriever
    - Alleviate the resource scarcity issue in low-resource languages

Language	Train Size	Dev Size	Test Size	Passage Length (Avg/Std/Mid)	#Keyphrases (Avg/Std/Mid)	Absent Kps%
AcademicMKP Dataset						
Chinese (ZH)	1,110	158	319	217/48/207	5/1/5	27.2%
Korean (KO)	774	110	222	115/31/111	4/1/4	37.7%
Total	1,884	268	541	171/57/155	4/1/4	31.3%
EcommerceMKP Dataset						
German (DE)	23,997	1,411	2,825	157/79/141	10/5/8	57.1%
Spanish (ES)	12,222	718	1,440	159/84/139	9/5/7	54.6%
French (FR)	16,986	998	2,000	163/84/144	9/5/8	63.0%
Italian (IT)	9,163	538	1,081	167/84/152	8/3/7	42.6%
Total	62,368	3,665	7,346	161/82/143	9/5/7	56.4%

Table 1: AcademicMKP & EcommerceMKP Dataset

Product Description (German): **Steiff** 113437 **Soft Cuddly Friends** Honey Teddybär, **grau**, 38 cm. Bereits der Name des **Soft Cuddly Friends** Honey Teddybär sagt es schon aus: der 38 cm große Freund mit seinem honigsüßen Lächeln begeistert alle Kinderherzen ...

(Translation in English): Steiff 113437 Soft Cuddly Friends Honey teddy bear, gray, 38 cm. The name of the Soft Cuddly Friends Honey Teddy bear already says it all: the 38 cm tall friend with his honey-sweet smile delights all children's hearts ...

Gold Keyphrases (German): *steiff kuscheltier*; *steiff teddy*; **soft cuddly friend**; **steiff**; *baer*; **grau**.

(Translation in English): steiff cuddly toy; steiff teddy; soft cuddly friend; steiff; bear; grey.

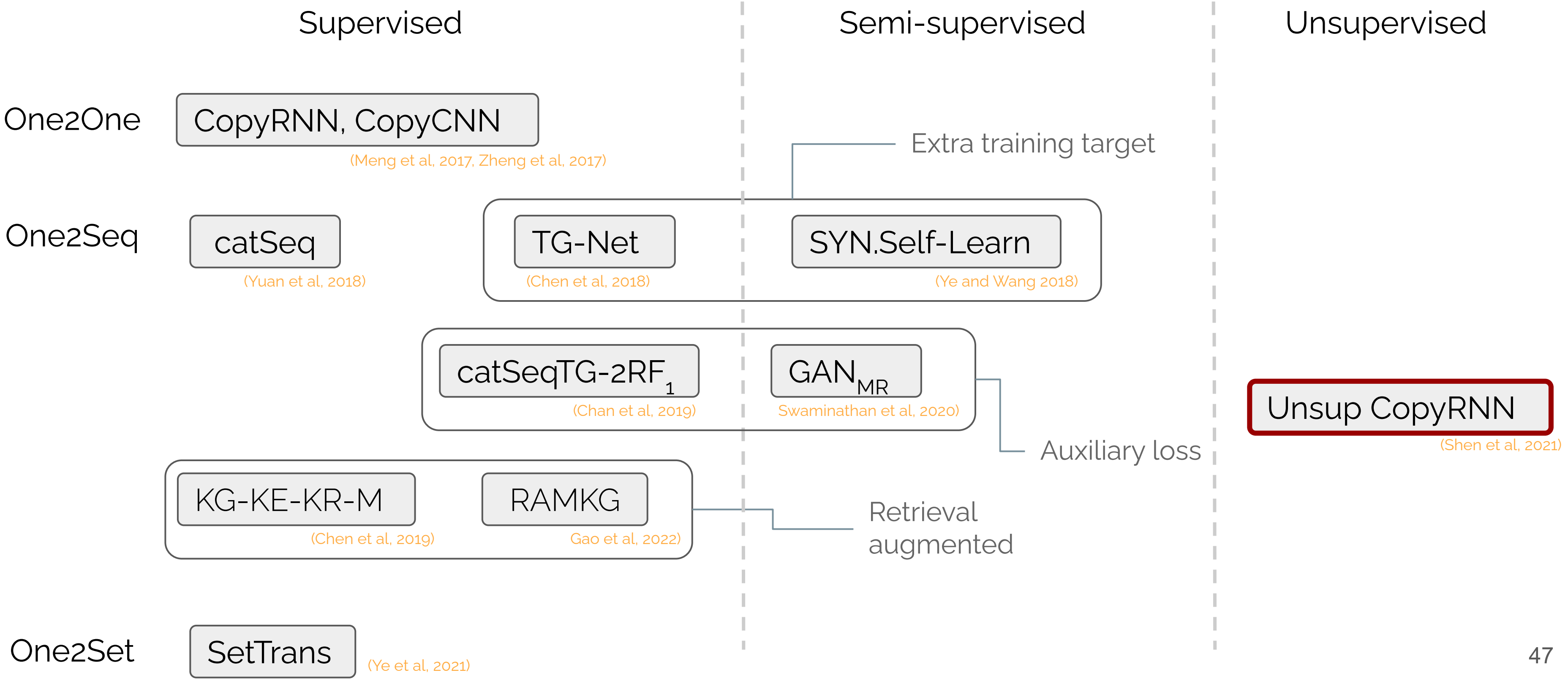
Retrieved English Keyphrases: steiff teddy bear; teddy bear; my first; grey; honey; sweetheart; steiff bear; pink; vintage; steiff stuffed animal; steiff; terry; soft; jimmy.

Predicted Keyphrases (German): *steiff kuscheltier*; *steiff teddy*; **soft cuddly friend**; **steiff**; *baer*; **grau**; *jimmy*.

(Translation in English): steiff cuddly toy; steiff teddy; soft cuddly friend; steiff; bear; grey; jimmy.

Yifan, Gao, *et al.* *"Retrieval-Augmented Multilingual Keyphrase Generation with Retriever-Generator Iterative Training."* NAACL findings. 2022.

Taxonomy of Generative Methods



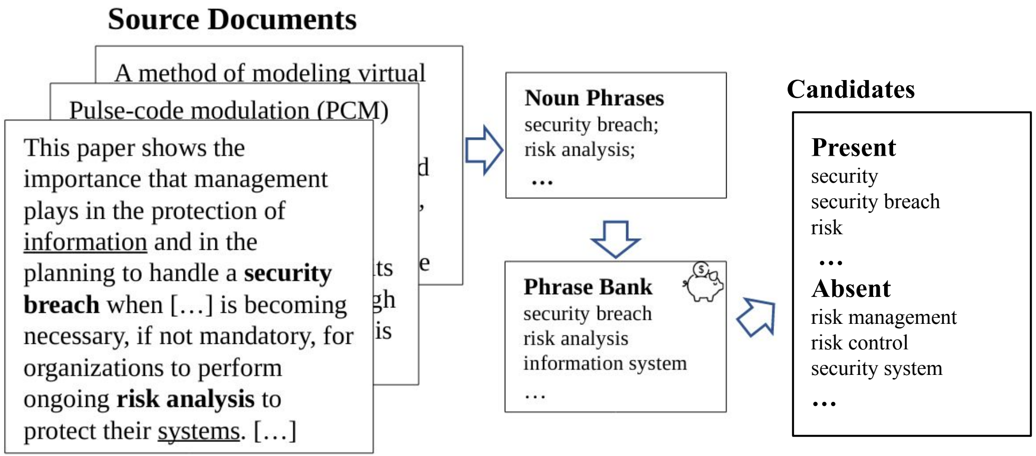
Can keyphrase generation be unsupervised?

- Why extractive methods can be unsupervised?
    - Selecting good spans from source texts is relatively easy
        - Extract candidates by n-grams, noun phrases etc.
    - Rank candidates with unsupervised scoring functions
        - Frequency-based
        - Graph-based
        - Semantic-based
        - ……

(Shen, *et al.* 2022). Unsupervised Deep Keyphrase Generation. AAAI.

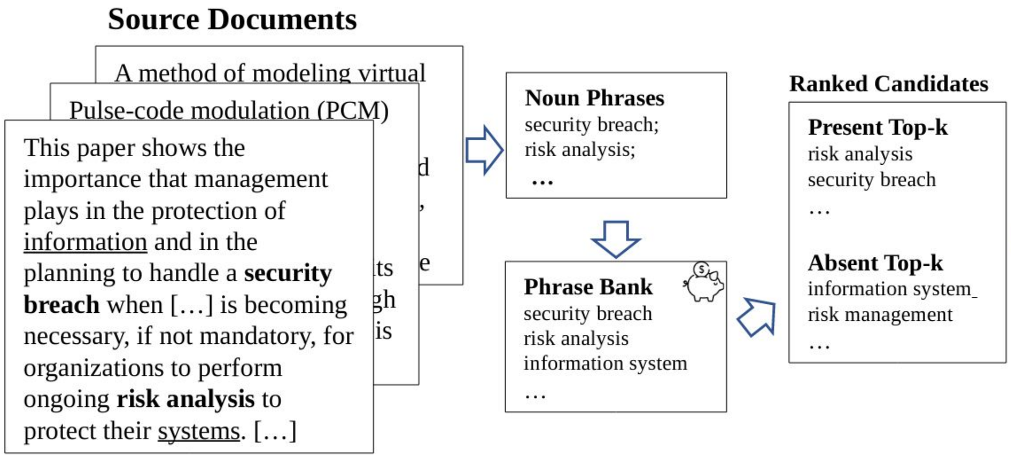
Unsupervised KPG

- 1. Construct synthetic text-keyphrase pairs w/o human annotation
  - Step 1.1: identify phrase candidates for each doc $\mathscr{d}$
    - Present candidates: noun phrases in $\mathscr{d}$ (with POS tagging)
    - Absent candidates: noun phrases from other docs $\bar{\mathscr{D}}$ if any word overlaps with $\mathscr{d}$



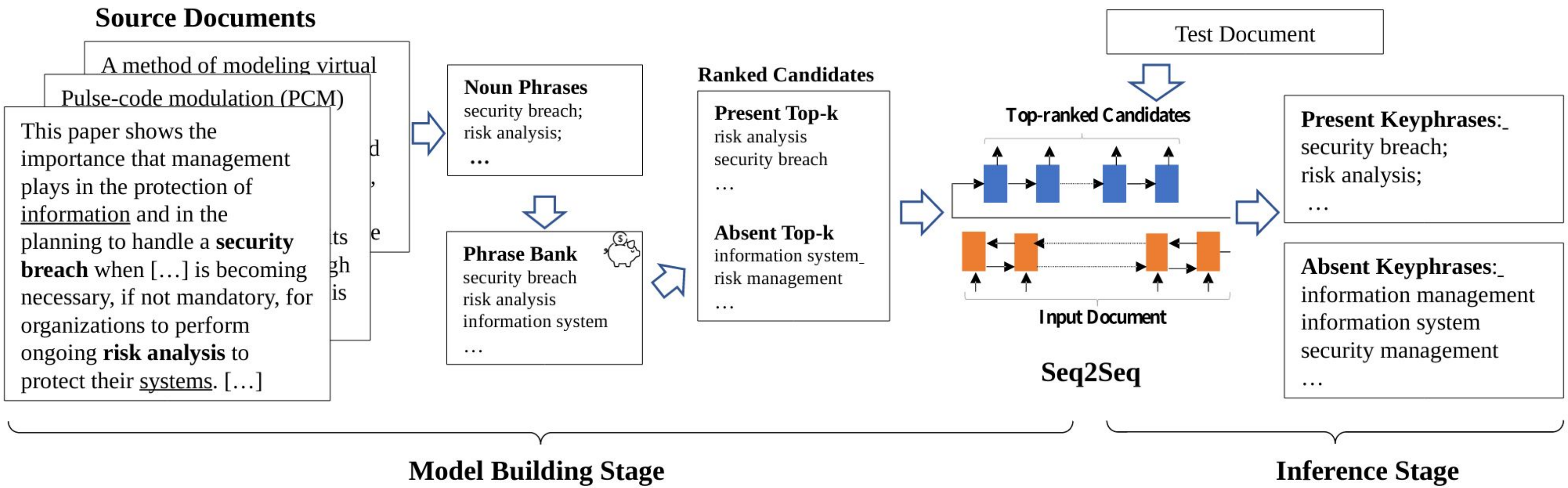
Unsupervised KPG

- 1. Construct synthetic text-keyphrase pairs w/o human annotation
  - Step 1.2: rank candidates by keyness
    - Lexical keyness: Tf-Idf
    - Semantic keyness: Similarity between a candidate and d by Doc2Vec
    - Fuse two scores: $\mathrm{RankScore}(\mathbf{x}, c) = \sqrt{\mathrm{Semantic}(\mathbf{x}, c)^{\lambda} \cdot \mathrm{Lexical}(\mathbf{x}, c)}$



Unsupervised KPG

- 2. Train a Seq2Seq model with synthetic pairs
    - Infuse keyphrase knowledge into the generation model



Unsupervised KPG

- Results on present keyphrase prediction

	Kp20K			Inspec			Krapivin			NUS			SemEval		
Model	**@5**	**@10**	**@$\mathcal{O}$**	**@5**	**@10**	**@$\mathcal{O}$**	**@5**	**@10**	**@$\mathcal{O}$**	**@5**	**@10**	**@$\mathcal{O}$**	**@5**	**@10**	**@$\mathcal{O}$**
TF-IDF	7.2	9.4	6.3	24.2	28.0	24.8	11.5	14.0	13.3	11.6	14.2	12.5	16.1	16.7	15.3
SingleRank	9.9	12.4	10.3	21.4	29.7	22.8	9.6	13.6	13.4	13.7	16.2	18.9	13.2	16.9	14.7
TextRank	18.1	15.1	14.1	26.3	27.9	26.0	14.8	13.9	13.0	18.7	19.5	19.9	<u>16.8</u>	18.3	18.1
ExpandRank	N/A	N/A	N/A	21.1	29.5	26.8	9.6	13.6	11.9	13.7	16.2	15.7	13.5	16.3	14.4
EmbedRank	15.5	15.6	15.8	29.5	34.4	<u>32.8</u>	13.1	13.8	13.9	10.3	13.4	14.7	10.8	14.5	13.9
AutoKeyGen	**23.4**	**24.6**	**23.8**	<u>30.3</u>	<u>34.5</u>	**33.1**	**17.1**	<u>15.5</u>	**15.8**	**21.8**	<u>23.3</u>	**23.7**	**18.7**	**24.0**	**22.7**
AutoKeyGen-OnlyBank	<u>22.9</u>	23.1	<u>23.1</u>	29.7	32.8	32.1	15.9	14.3	14.2	20.7	21.8	22.3	16.3	20.9	20.4
AutoKeyGen-OnlyEmbed	21.2	22.9	21.8	29.7	**34.8**	32.7	15.9	**16.4**	14.3	20.4	21.3	22.6	15.3	16.5	15.9
AutoKeyGen-CopyRNN	22.7	<u>24.2</u>	**23.8**	**30.5**	33.2	32.7	<u>16.6</u>	15.1	<u>14.7</u>	<u>21.6</u>	**22.4**	<u>22.7</u>	**18.7**	<u>22.3</u>	<u>21.4</u>
Supervised-CopyRNN	33.1	27.9	35.3	28.5	32.5	33.7	32.0	27.0	35.5	40.2	35.9	43.4	32.9	34.6	35.2

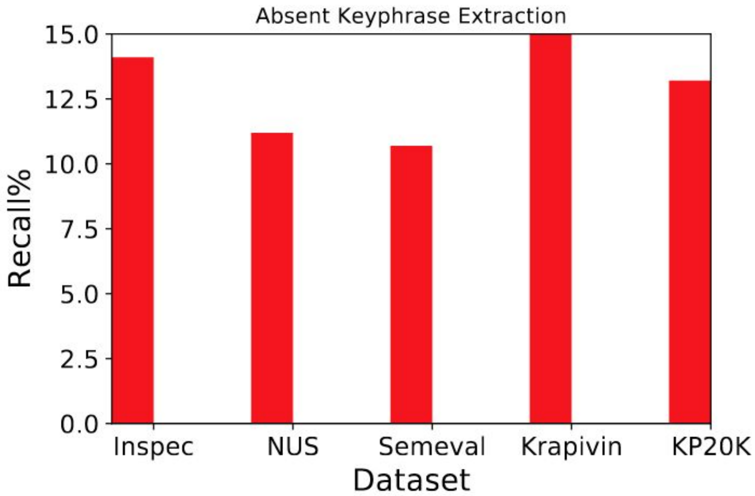
Unsupervised KPG

- Results on absent keyphrase prediction

	Kp20K		Inspec		Krapivin		NUS		SemEval	
Model	**R@10**	**R@20**	**R@10**	**R@20**	**R@10**	**R@20**	**R@10**	**R@20**	**R@10**	**R@20**
Other Unsupervised Methods	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
ExpandRank	N/A	N/A	0.02	0.05	0.01	0.015	0.005	0.04	0	0.004
AutoKeyGen	**2.3**	**2.5**	**1.7**	**2.1**	**3.3**	**5.4**	**2.4**	**3.2**	**1.0**	**1.1**
AutoKeyGen-OnlyBank	1.8	2.2	1.5	1.7	3.1	4.1	2.1	2.6	0.7	0.9
AutoKeyGen-OnlyEmbed	1.9	2.3	1.4	1.8	3.0	4.5	2.1	2.7	0.9	0.9
AutoKeyGen-CopyRNN	1.8	2.0	1.6	1.9	3.1	4.7	1.9	2.8	**1.0**	**1.1**
Supervised-CopyRNN	11.5	14.0	5.1	6.8	11.6	14.2	7.8	10.0	4.9	5.7

Unsupervised KPG

- Recall of absent keyphrases using all phrase candidates in corpus
    - Upper bound score of current method on absent keyphrases
    - Some absent phrases are missed by POS tagging



Taxonomy of Methods

Traditional methods | Neural-based methods

Extraction

- statistical (e.g. TF.IDF)
- graph-based (e.g. TextRank)
- supervised feature-based (e.g. Kea)
- sequence labeling: CRF, Bi-LSTM-CRF (Das Gollapalli et al., 2017) (Alzaidy et al. 2019)
- embedding-based (e.g. Key2Vec) (Mahata et al., 2018)

Generation

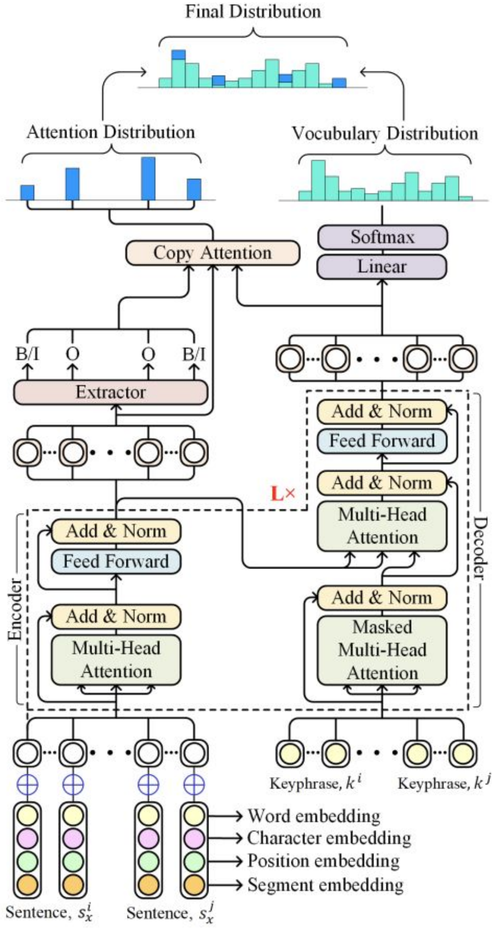
- resource-based (e.g. Kea++) (Medelyan and Witten, 2006)
- sequence-to-sequence (e.g. CopyRNN)

Extraction + **Generation**

(Medelyan and Witten, 2006) Thesaurus based automatic keyphrase indexing. JCDL.
(Das Gollapalli et al., 2017) Incorporating expert knowledge into keyphrase extraction. AAAI.
(Mahata et al., 2018) Key2Vec: Automatic Ranked Keyphrase Extraction from Scientific Articles using Phrase Embeddings. NAACL.
(Alzaidy et al. 2019) Bi-LSTM-CRF Sequence Labeling for Keyphrase Extraction from Scholarly Documents. WWW.

Unifying Extraction & Generation

- Combining extraction & generation as multi-tasking
  - Generator takes the hint of present phrases predicted by extractor
- Related work
  - KG-KE-KR-M, Chen et al., NAACL 2019
  - SEG-Net, Ahmad, Wasi, et al., ACL 2021
  - UniKeyphrase, Wu et al., ACL Finding 2021
  - BERT-PKE & BERT-AKG, Liu et al., Arxiv 2020



(Chen, 2019) An Integrated Approach for Keyphrase Generation via Exploring the Power of Retrieval and Extraction NAACL.
(Ahmad Wasi et al., 2021) Select, extract and generate: Neural keyphrase generation with layer-wise coverage attention. ACL.
(Wu et al., 2021) UniKeyphrase: A Unified Extraction and Generation Framework for Keyphrase Prediction. ACL Finding.
(Liu et al. 2020) Keyphrase prediction with pre-trained language model. arXiv

ECIR 2022

Unifying Extraction & Generation

- SEG-Net

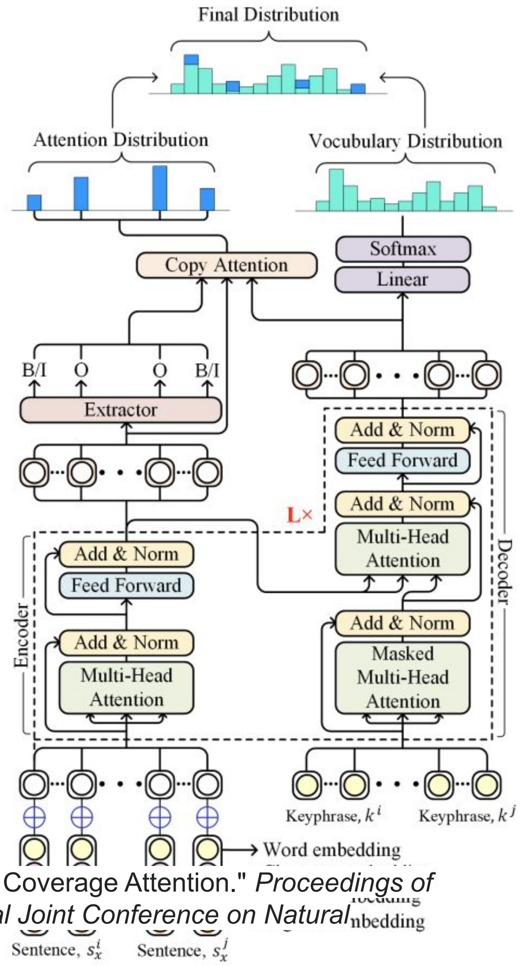
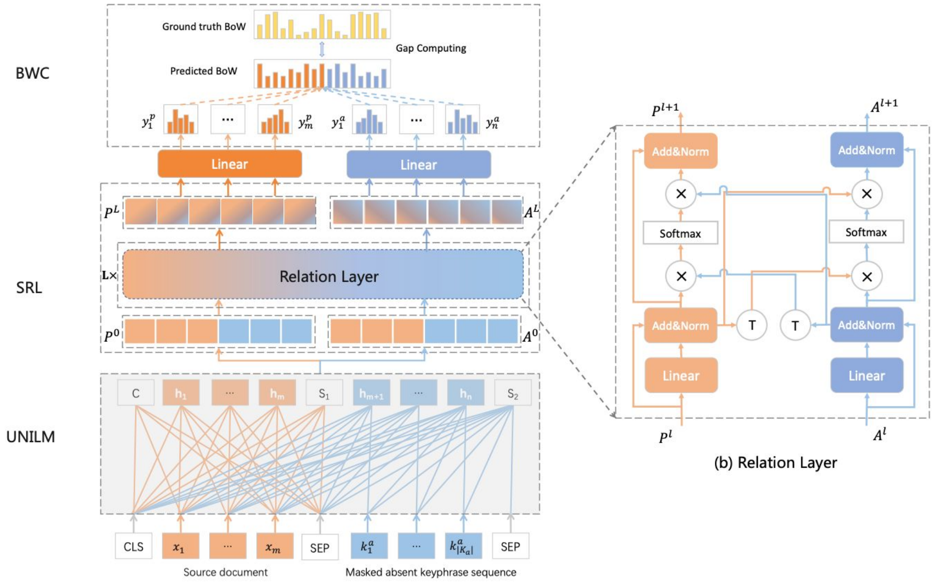


Figure 2: Overview of the Extractor-Generator module of SEG-Net. The major components are encoder, extractor, and decoder. The encoder encodes the salient sentences of the input document. The extractor predicts ... [illegible] ... nerates the absent keyphrases word by word.

Ahmad, Wasi, et al. "Select, Extract and Generate: Neural Keyphrase Generation with Layer-wise Coverage Attention." *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. 2021.

Unifying Extraction & Generation

- UniKeyphrase

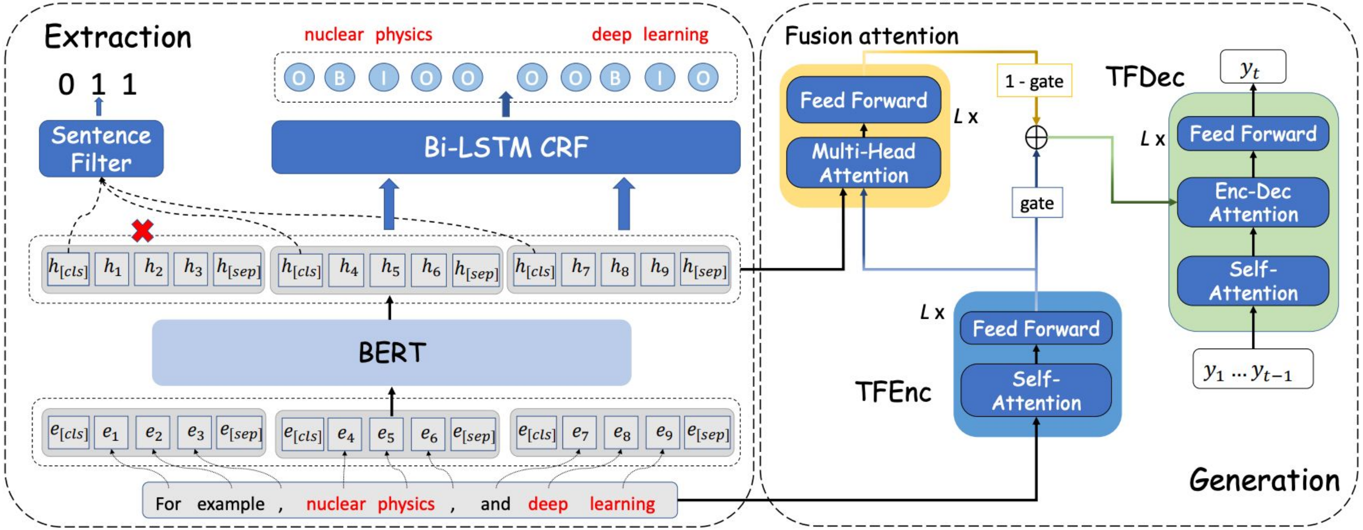


(a) UniKeyphrase
(b) Relation Layer

Wu, Huanqin, et al. "UniKeyphrase: A Unified Extraction and Generation Framework for Keyphrase Prediction." *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*. 2021.

Unifying Extraction & Generation

- BERT-PKE & BERT-AKG



Liu, Rui, Zheng Lin, and Weiping Wang. "Keyphrase prediction with pre-trained language model." *arXiv preprint arXiv:2004.10462* (2020).

Conclusion - Neural Keyphrasification

- Learn to predict keyphrases in a data-driven manner
  - No manual feature engineering
  - Outperform classic methods by large margins
- Challenges
  - Data-hungry
  - Weak generalizability across domains
  - Quality of generated absent phrases

Outline of Part II

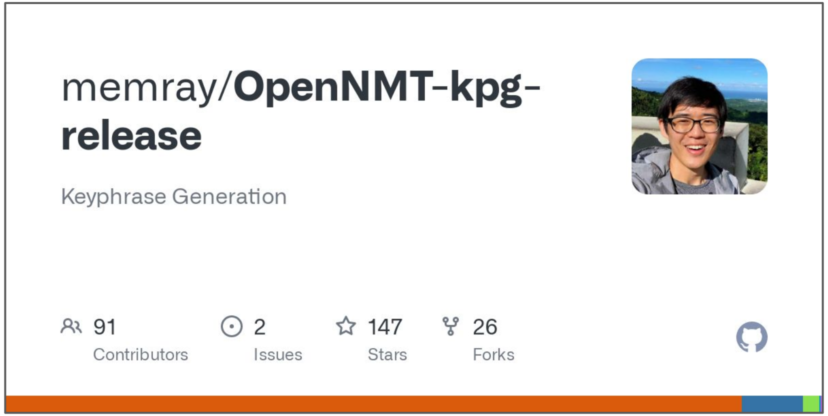
Part I - Neural Keyphrase Extraction (Debanjan)

Part II - Neural Keyphrase Generation (Rui)

Part III - Introduction to OpenNMT-kpg and DLKP

Intro to OpenNMT-kpg

- A PyTorch package for keyphrase generation based on OpenNMT



Intro to OpenNMT-kpg

- Features
  - Configure job in yml files and start training in one line

    python train.py -config config/train/transformer-presabs-kptimes.yml

```yaml
train_from: /zfs1/hdaqing/rum20/kp/fairseq-kpg/exps/kp/transformer_presabs_kptimes/ckpts/checkpoint_step_160000.pt

### Exp meta
exp: transformer-presabs-kptimes
exp_dir: /zfs1/hdaqing/rum20/kp/fairseq-kpg/exps/kp/transformer_presabs_kptimes
save_model: /zfs1/hdaqing/rum20/kp/fairseq-kpg/exps/kp/transformer_presabs_kptimes/ckpts/checkpoint
log_file: /zfs1/hdaqing/rum20/kp/fairseq-kpg/exps/kp/transformer_presabs_kptimes/log.txt
wandb_project: transfer_kp

### Data opts:
data_type: keyphrase
pretrained_tokenizer: true # using roberta_tokenize_kpg transform
data_format: jsonl
save_data: /zfs1/hdaqing/rum20/kp/data/kp/generated/dynamic.ex0
overwrite: False
cache_dir: /zfs1/hdaqing/rum20/kp/data/kp/cache/

src_seq_length_trunc: 512
tgt_seq_length_trunc: 128
shuffle_shards: false
data:
    corpus_1:
        path_src: /zfs1/hdaqing/rum20/kp/data/kp/json/kptimes/train.json
        type: keyphrase
        transforms: [keyphrase, roberta_tokenize_kpg]

### Transform related opts:
#### Keyphrase specific
kp_concat_type: pres_abs
#### Subword and vocab
src_subword_model: roberta_tokenize
src_vocab: /zfs1/hdaqing/rum20/kp/data/kp/hf_vocab/roberta-base-kp/vocab.json
share_vocab: True
bpe_dropout: 0.0
```

Intro to OpenNMT-kpg

- Features
  - Data preprocessing on-the-fly
    - You can pipeline the data processing like a charm

```yaml
data:
    corpus_1:
        path_src: /zfs1/hdaqing/rum20/kp/data/kp/json/kptimes/train.json
        type: keyphrase
        transforms: [keyphrase, roberta_tokenize_kpg]
```

```python
def apply(self, example, is_train=False, stats=None, **kwargs):
    """
    Source text: concatenating title and body text.
    Target text: concatenating phrases according to the given phrase order.
    """
    if self.use_given_inputs and 'src' in example and 'tgt' in example and example['src'] and example['tgt']:
        print('WARNING: using src and tgt that are directly given rather than processed on-the-fly.\n '
              'This is only designated to ease the out-of-the-box inference. Ensure this behavior is wanted.')
        return example

    dataset_type = self.infer_dataset_type(example)
    src_tokens, tgt_tokens, src_str, tgt_str = self.kpdict_parse_fn(example, self.kp_concat_type, dataset_type=dataset_type)
    if self.return_tokens:
        example['src'] = src_tokens
        example['tgt'] = tgt_tokens
    else:
        example['src'] = src_str
        example['tgt'] = tgt_str

    example['src_str'] = src_str
    example['tgt_str'] = tgt_str

    return example
```

Intro to OpenNMT-kpg

- Features
  - Easy keyphrase inference with huggingface datasets and pretrained models
    - The inference example is located at /onmt/keyphrase/kpg_example_hfdatasets.py

```
if __name__ == '__main__':
    # load dataset
    dataset_name = 'midas/inspec'
    kp_dataset = datasets.load_dataset(dataset_name, name='raw', split='test')

    # load configs
    config_path = '/zfs1/hdaqing/rum20/kp/OpenNMT-kpg-transfer/config/transfer_kp/infer/keyphrase-one2seq-controlled.yml'
    ckpt_path = '/zfs1/pbrusilovsky/rum20/kp/openNMT-kpg-release-ckpt/wiki-pretrained/bart-wiki-step40k-bs256.checkpoint_step_40000.pt'
    parser = _get_parser()
    opt = parser.parse_args('-config %s' % (config_path))
```

```
========================================================
Summary of scores on midas/inspec
    count-num_gold  =       12.5000
    count-num_present_gold  =       10.6667
    count-num_absent_gold   =       1.8333
    count-num_pred  =       27.6667
    count-num_valid_pred    =       23.1667
    count-num_present_pred  =       12.3333
    count-num_present_valid_pred    =       8.8333
    count-num_absent_valid_pred =       14.3333
    all_exact-f_score@5 =       0.1751
    all_exact-f_score@10    =       0.1637
    all_exact-f_score@k =       0.1694
    present_exact-f_score@5 =       0.1900
    present_exact-f_score@10    =       0.1946
    present_exact-f_score@k =       0.2067
    absent_exact-f_score@50 =       0.0000
    absent_exact-f_score@M  =       0.0000

========================================================
```

Intro to DLKP

midas-research/
dlkp

A deep learning library for identifying keyphrases from text

2	7	2	0
Contributors	Issues	Stars	Forks

From Fundamentals to Recent Advances
A Tutorial on Keyphrasification

All materials available at
https://keyphrasification.github.io/