Salience Rank

Efficient Unsupervised Keyphrase Extraction with Topic Modeling

Weiwei Cheng

Amazon Development Center Germany

Berlin Germany

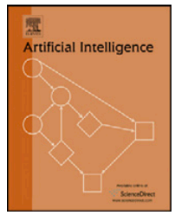
joint work with Nedelina Teneva

Artificial Intelligence 172 (2008) 1897–1916

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint



Label ranking by learning pairwise preferences

Eyke Hüllermeier [a,*], Johannes Fürnkranz [b], Weiwei Cheng [a], Klaus Brinker [a]

ARTICLE INFO

Article history:
Received 21 January 2008
Received in revised form 14 July 2008
Accepted 8 August 2008
Available online 15 August 2008

Keywords:
Preference learning
Ranking
Pairwise classification
Constraint classification

ABSTRACT

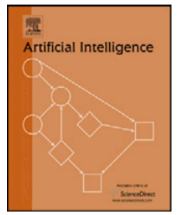
Preference learning is an emerging topic that appears in different guises in the recent literature. This work focuses on a particular learning scenario called label ranking, where the problem is to learn a mapping from instances to rankings over a finite number of labels. Our approach for learning such a mapping, called ranking by pairwise comparison (RPC), first induces a binary preference relation from suitable training data using a natural extension of pairwise classification. A ranking is then derived from the preference relation thus obtained by means of a ranking procedure, whereby different ranking methods can be used for minimizing different loss functions. In particular, we show that a simple (weighted) voting strategy minimizes risk with respect to the well-known Spearman rank correlation. We compare RPC to existing label ranking methods, which are based on scoring individual labels instead of comparing pairs of labels. Both empirically and theoretically, it is shown that RPC is superior in terms of computational efficiency, and at least competitive in terms of accuracy.

Artificial Intelligence 172 (2008) 1897–1916

Contents lists available at ScienceDirect

Artificial Intelligence

www.elsevier.com/locate/artint



Label ranking by learning pairwise preferences

Eyke Hüllermeier [a,*], Johannes Fürnkranz [b], Weiwei Cheng [a], Klaus Brinker [a]

ARTICLE INFO

Article history:
Received 21 January 2008
Received in revised form 14 July 2008
Accepted 8 August 2008
Available online 15 August 2008

Keywords:
Preference learning
Ranking
Pairwise classification
Constraint classification

ABSTRACT

Preference learning is an emerging topic that appears in different guises in the recent literature. This work focuses on a particular learning scenario called label ranking, where the problem is to learn a mapping from instances to rankings over a finite number of labels. Our approach for learning such a mapping, called ranking by pairwise comparison (RPC), first induces a binary preference relation from suitable training data using a natural extension of pairwise classification. A ranking is then derived from the preference relation thus obtained by means of a ranking procedure, whereby different ranking methods can be used for minimizing different loss functions. In particular, we show that a simple (weighted) voting strategy minimizes risk with respect to the well-known Spearman rank correlation. We compare RPC to existing label ranking methods, which are based on scoring individual labels instead of comparing pairs of labels. Both empirically and theoretically, it is shown that RPC is superior in terms of computational efficiency, and at least competitive in terms of accuracy.

Adina Apartment Hotel Berlin Checkpoint Charlie

2,702 reviews | #14 of 634 Hotels in Berlin | Certificate of Excellence

Show Prices

00 49 2302 4399016 | Hotel website | E-mail hotel | Krausenstrasse 35 - 36, 10117 Berlin, Germany

2,702 Reviews from our TripAdvisor Community

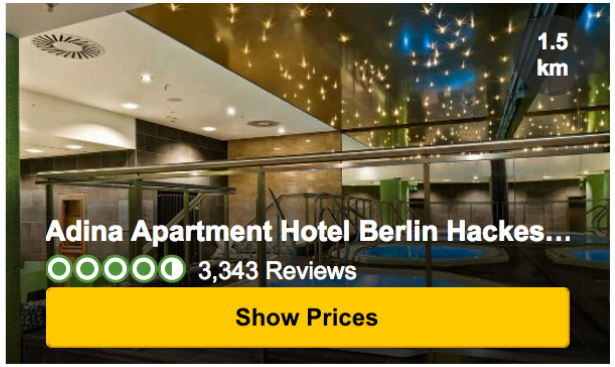
Write a Review | Add Photo

Read reviews that mention:

Search reviews

All reviews | checkpoint charlie | washing machine and tumble dryer | lidl supermarket | pool and jacuzzi | separate bedroom | full kitchen | self cater | great hotel | small kitchen | living area | building site | building work | within walking distance | minute walk | check point charlie | jewish museum | u bahn | would definitely stay here again | central location | great facilities

Explore similar hotels



tripadvisor IRELAND | Adina Apartment Hotel Berlin Checkpoint Charlie Reviews | Review | JOIN | LOG IN | € |

Berlin | Hotels | Flights | Holiday Rentals | Restaurants | Things to Do | Forum | Best of 2017 | More

Find: Hotels | Near: Berlin, Germany | Search

Europe › Germany › Berlin › Berlin Hotels

Adina Apartment Hotel Berlin Checkpoint Charlie

2,702 reviews | #14 of 634 Hotels in Berlin | Certificate of Excellence

Show Prices

00 49 2302 4399016 | Hotel website | E-mail hotel | Krausenstrasse 35 - 36, 10117 Berlin, Germany

2,702 Reviews from our TripAdvisor Community

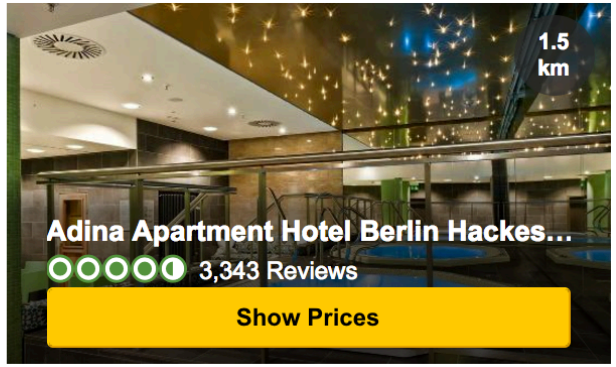
Write a Review | Add Photo

Read reviews that mention:

Search reviews

All reviews | checkpoint charlie | washing machine and tumble dryer | lidl supermarket | pool and jacuzzi | separate bedroom | full kitchen | self cater | great hotel | small kitchen | living area | building site | building work | within walking distance | minute walk | check point charlie | jewish museum | u bahn | would definitely stay here again | central location | great facilities

Explore similar hotels



Keyphrase Extraction

Keyphrase extraction aims to find a collection of phrases in a document that provides a concise summary of the text content.

- **Inputs**: a text document
- **Outputs**: a set/ranking of phrases

Keyphrase Extraction

Keyphrase extraction aims to find a collection of phrases in a document that provides a concise summary of the text content.

- **Inputs**: a text document
- **Outputs**: a set/ranking of phrases
- **Evaluation** is done by comparing to human annotated keyphrases via measures such as *precision*, *recall*, *F score*, etc.

Keyphrase Extraction

An automatic keyphrase extraction system typically operates in 2 steps:

1. Extract a list of phrases as **candidate phrases** with some heuristics.

2. Select keyphrases from these candidates with **supervised** or **unsupervised** approaches.

****
****
****
****
****

Keyphrase Extraction

An automatic keyphrase extraction system typically operates in 2 steps:

1. Extract a list of phrases as **candidate phrases** with some heuristics.

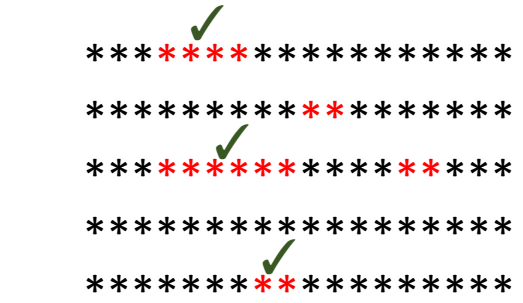
2. Select keyphrases from these candidates with **supervised** or **unsupervised** approaches.

Keyphrase Extraction

An automatic keyphrase extraction system typically operates in 2 steps:

1. Extract a list of phrases as **candidate phrases** with some heuristics.

2. Select keyphrases from these candidates with **supervised** or **unsupervised** approaches.



Keyphrase Extraction

An automatic keyphrase extraction system typically operates in 2 steps:

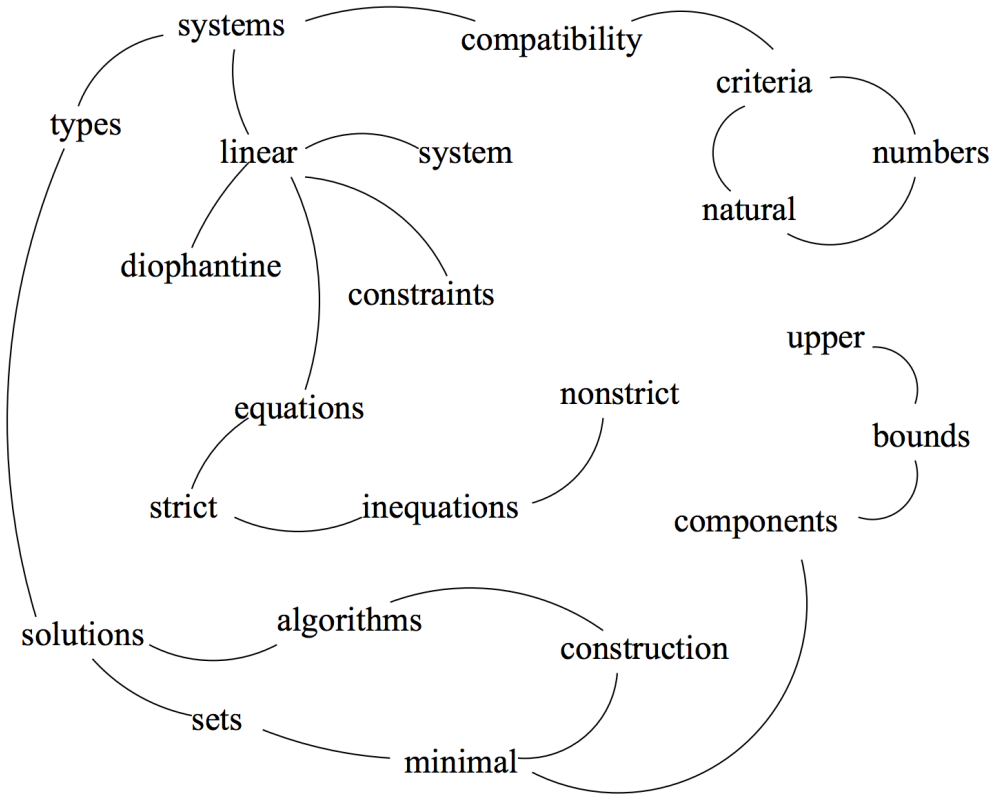
1. Extract a list of phrases as **candidate phrases** with some heuristics.
   - Noun phrases with `(adjective)*(noun)+`
   - Phrases that don't contain predefined stopwords
   - etc.
2. Select keyphrases from these candidates with **supervised** or **unsupervised** approaches.

Keyphrase Extraction

An automatic keyphrase extraction system typically operates in 2 steps:

1. Extract a list of phrases as **candidate phrases** with some heuristics.
   - Noun phrases with `(adjective)*(noun)+`
   - Phrases that don't contain predefined stopwords
   - etc.
2. Select keyphrases from these candidates with **supervised** or **unsupervised** approaches.
   - Supervised: binary classification (Frank et al. 1999), pairwise ranking (Jiang et al. 2009)
   - Unsupervised: graph-based ranking (Mihalcea & Tarau, 2004), topic-based clustering (Grineva et al., 2009), language modeling (Tomokiyo & Hurst, 2003)

Graph-Based Ranking



Intuition:

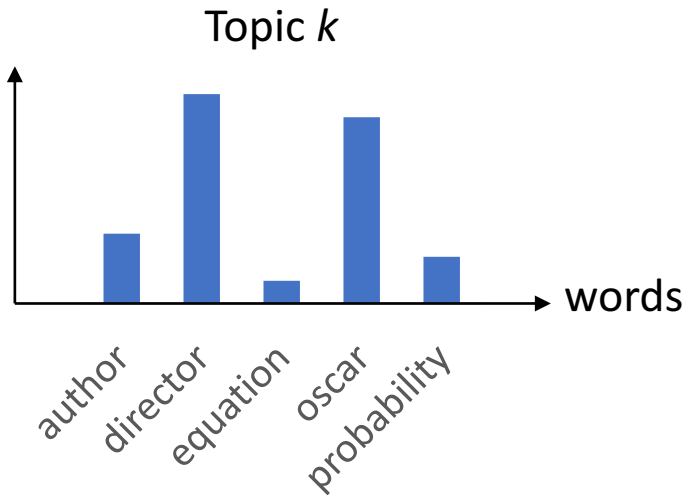
A candidate keyphrase is important if it is related to other candidates, which in turn also have high importance.

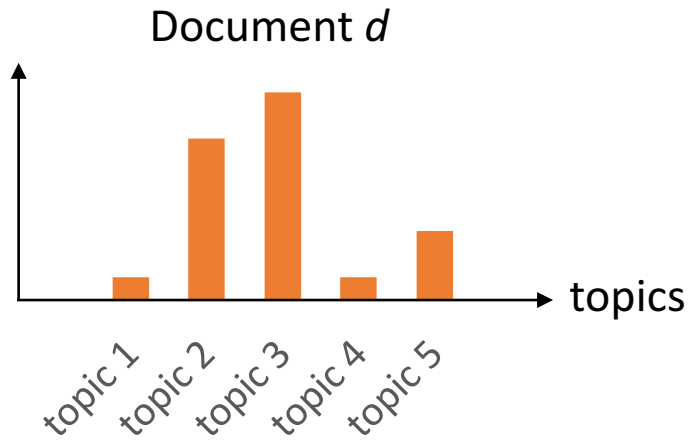
Procedure: e.g., (Mihalcea & Tarau, 2004)

1. Build a word graph from the input document
2. Perform random walk (e.g., PageRank) to obtain word scores
3. Select keyphrases with word scores

Topical PageRank (Liu et al., 2010)

- Main idea: Use latent topic distribution inferred by **LDA**, latent Dirichlet allocation (Blei et al., 2003), to guide the random walk on the word graph.
- In LDA, a topic is a distribution over the vocabulary; each document is viewed as a mixture of topics.





Topical PageRank (Liu et al., 2010)

- Given a word graph $G = (W, E)$, where vertices represent words and an edge $e(w_i, w_j)$ indicates relatedness between w_i and w_j, the score of each word w_i under topic $t \in T$ is determined by the random walk

$$R_t(w_i) = \lambda \sum_{j:\, w_j \to w_i} \frac{e(w_i, w_j)}{Out(w_j)} R_t(w_j) + (1 - \lambda)\, p(t \mid w_i)\,, \qquad (1)$$

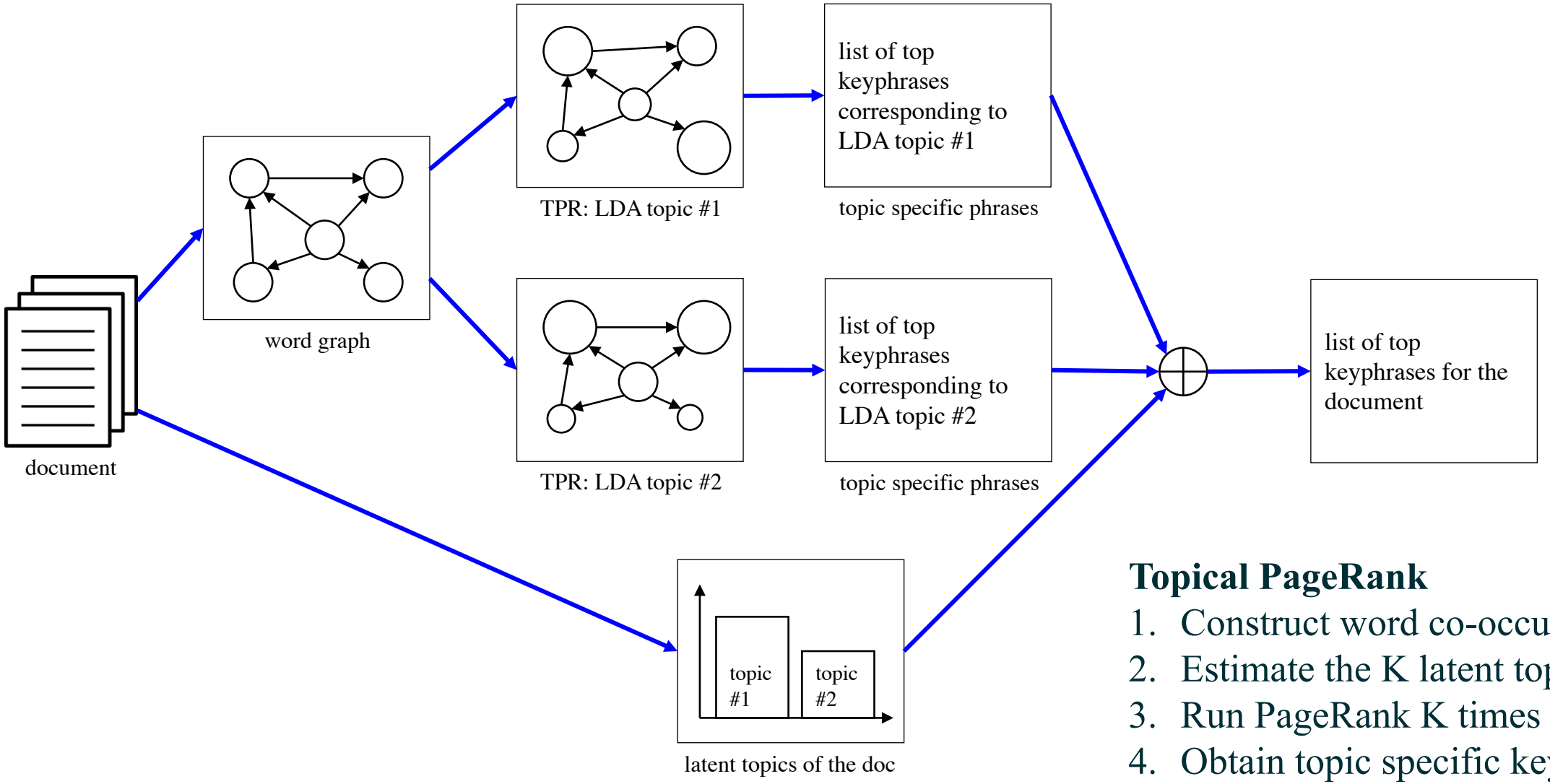
where $Out(w_i) = \sum_{i:\, w_i \to w_j} e(w_i, w_j)$ is the outdegree of vertex w_i, and $p(t \mid w_i)$, derived from LDA, is a topic specific jump probability of w_i.

Topical PageRank (Liu et al., 2010)

- Given a word graph $G = (W, E)$, where vertices represent words and an edge $e(w_i, w_j)$ indicates relatedness between w_i and w_j, the score of each word w_i under topic $t \in T$ is determined by the random walk

$$R_t(w_i) = \lambda \sum_{j:\, w_j \to w_i} \frac{e(w_i, w_j)}{Out(w_j)} R_t(w_j) + (1 - \lambda)\, p(t \mid w_i)\,, \qquad (1)$$

  where $Out(w_i) = \sum_{i:\, w_i \to w_j} e(w_i, w_j)$ is the outdegree of vertex w_i, and $p(t \mid w_i)$, derived from LDA, is a topic specific jump probability of w_i.
- Then for topic t, we obtain keyphrase scores $R_t(\text{phrase}) = \sum_{w_i \in \text{phrase}} R_t(w_i)$.
- The final keyphrase scores are given by $R(\text{phrase}) = \sum_{t \in \text{T}} R_t(\text{phrase})\, p(t \mid \text{d})$.



Topical PageRank

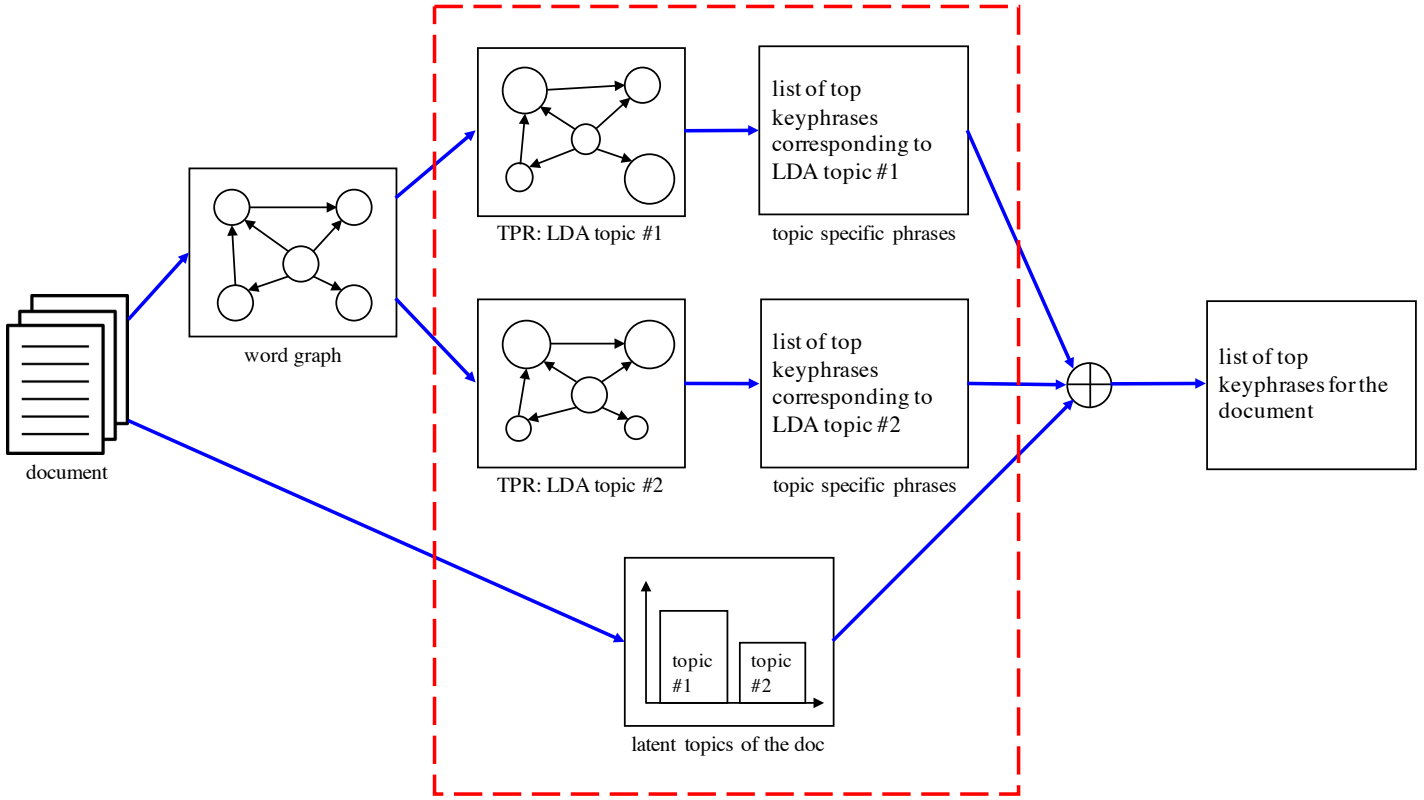
1. Construct word co-occurrence graph
2. Estimate the K latent topics with LDA
3. Run PageRank K times with (1)
4. Obtain topic specific keyphrase scores
5. Obtain the overall keyphrase scores

Our Contribution – Salience Rank

- **Performance**: While still exploiting the structure information derived by LDA, we run PageRank once instead of K times and achieve similar keyphrase quality.

Our Contribution – Salience Rank

- **Performance**: While still exploiting the structure information derived by LDA, we run PageRank once instead of K times and achieve similar keyphrase quality.



Our Contribution – Salience Rank

- **Performance**: While still exploiting the structure information derived by LDA, we run PageRank once instead of K times and achieve similar keyphrase quality.
- **Configurability**: Users can balance *topic specificity* and *corpus specificity* of the extracted keyphrases and can tune the results according to particular use cases.

Our Contribution – Salience Rank

- **Performance**: While still exploiting the structure information derived by LDA, we run PageRank once instead of K times and achieve similar keyphrase quality.
- **Configurability**: Users can balance *topic specificity* and *corpus specificity* of the extracted keyphrases and can tune the results according to particular use cases.
  - On one hand, we aim to extract keyphrases that are relevant to specific topics;
  - On the other hand, the extracted keyphrases as a whole should have a good coverage of the major topics in the document.
  - It is often useful to control the balance between these two competing principles.

Our Contribution – Salience Rank

- **Definition.** The *topic specificity* of a word w is

$$TS(w) = \sum_{t \in T} p(t \mid w) \log \frac{p(t \mid w)}{p(t)}$$

$$= KL(p(t \mid w) \,||\, p(t))$$

Our Contribution – Salience Rank

- **Definition.** The *topic specificity* of a word w is

$$TS(w) = \sum_{t \in T} p(t \mid w) \log \frac{p(t \mid w)}{p(t)}$$

$$= KL(p(t \mid w) \,||\, p(t))$$

- **Example.** Suppose $p(t_1) = p(t_2) = 0.5$, we consider three words with

$$\begin{aligned} p(t_1|w_1) &= 0.9, & p(t_2|w_1) &= 0.1 \\ p(t_1|w_2) &= 0.7, & p(t_2|w_2) &= 0.3 \\ p(t_1|w_3) &= 0.5, & p(t_2|w_3) &= 0.5 \end{aligned}$$

We have $TS(w_1) = 0.53$, $TS(w_2) = 0.12$, and $TS(w_3) = 0$.

Our Contribution – Salience Rank

- **Definition.** The *topic specificity* of a word w is

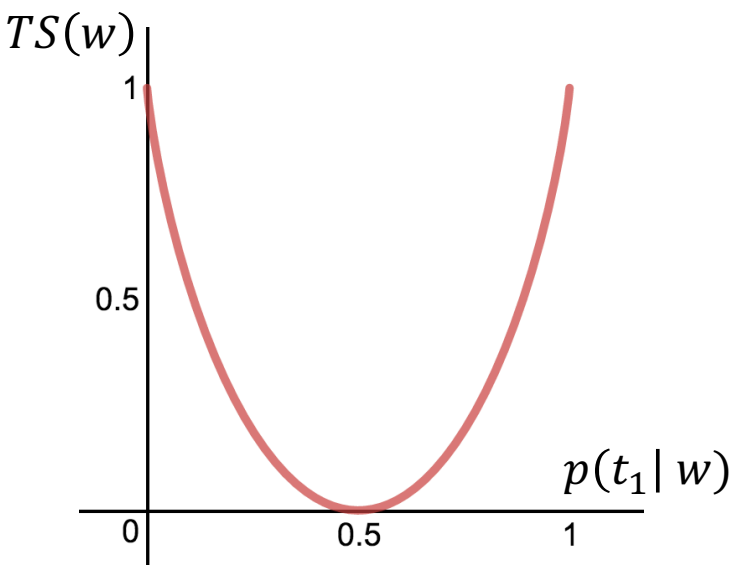
$$TS(w) = \sum_{t \in T} p(t \mid w) \log \frac{p(t \mid w)}{p(t)}$$

$$= KL(p(t \mid w) \,||\, p(t))$$

- **Example.** Suppose $p(t_1) = p(t_2) = 0.5$, we consider three words with

$$p(t_1|w_1) = 0.9, \quad p(t_2|w_1) = 0.1$$
$$p(t_1|w_2) = 0.7, \quad p(t_2|w_2) = 0.3$$
$$p(t_1|w_3) = 0.5, \quad p(t_2|w_3) = 0.5$$

We have $TS(w_1) = 0.53$, $TS(w_2) = 0.12$, and $TS(w_3) = 0$.



Our Contribution – Salience Rank

- **Definition.** The *topic specificity* of a word w is

$$TS(w) = \sum_{t \in T} p(t \mid w) \log \frac{p(t \mid w)}{p(t)}$$

- **Definition.** The *corpus specificity* of a word w is

$$CS(w) = p(w \mid \text{corpus})$$

Our Contribution – Salience Rank

- **Definition.** The *topic specificity* of a word w is

$$TS(w) = \sum_{t \in T} p(t \mid w) \log \frac{p(t \mid w)}{p(t)}$$

- **Definition.** The *corpus specificity* of a word w is

$$CS(w) = p(w \mid \text{corpus})$$

- **Definition.** The *salience* of a word w is

$$S(w) = (1 - \alpha)\, CS(w) + \alpha\, TS(w)$$

where α is the tradeoff parameter balancing corpus and topic specificity of w.

We use normalized $TS(w) \in [0,1]$ when defining $S(w)$.

Our Contribution – Salience Rank

- **Definition.** The *topic specificity* of a word w is

$$TS(w) = \sum_{t \in T} p(t \mid w) \log \frac{p(t \mid w)}{p(t)}$$

- **Definition.** The *corpus specificity* of a word w is

$$CS(w) = p(w \mid \text{corpus})$$

high, when word occurs often in the corpus

high, when word is less shared across topics

- **Definition.** The *salience* of a word w is

$$S(w) = (1 - \alpha)\, CS(w) + \alpha\, TS(w)$$

where α is the tradeoff parameter balancing corpus and topic specificity of w.

Our Contribution – Salience Rank

Our random walk:

$$R(w_i) = \lambda \sum_{j:\, w_j \to w_i} \frac{e(w_i, w_j)}{Out(w_j)} R(w_j) + (1-\lambda)\, S(w_i)$$

Comparing to (1) in TPR, PageRank needs to be run **only once**.

Empirical Evaluation – Performance

dataset	algorithm	precision	recall	F score
500news	TPR	0.254	0.222	0.229 (±0.010)
	SR	0.253	0.222	0.229 (±0.010)
Inspec	TPR	0.225	0.255	0.227 (±0.007)
	SR	0.265	0.298	0.266 (±0.007)

- While computationally more efficient, Salience Rank obtains comparable or better keyphrases on benchmark data.
- More details are in the paper, including comparisons to other approaches, parameter settings, etc.

Empirical Evaluation – Configurability

	α	precision	recall	F score	
high TS	1.0	0.247	0.216	0.223 (±0.011)	500news
	0.7	0.248	0.216	0.223 (±0.011)	
	0.4	0.248	0.217	0.224 (±0.011)	
	0.1	0.254	0.222	0.229 (±0.010)	
high CS	0.0	0.248	0.217	0.224 (±0.011)	

- The tradeoff between topic and corpus specificity has a considerable impact on the performance measures.

Empirical Evaluation – Configurability

Results of Salience Rank on one Inspec abstract with extreme values of α:

highest CS

Unique top keyphrases when $\alpha = 0$	**Unique top keyphrases when $\alpha = 1$**
classical mathematical formalization	individual interests
preferences	group interests
theory	artificial social systems
options	individual rationality
function	conditional preference relationships
multiple agent settings	Neumann-Morgenstern theory

highest TS

Intuitively, the left is good for a layman and the right is good for an expert.

Conclusions & Possible Applications

We proposed an unsupervised keyphrase extraction algorithm, **Salience Rank**, that improves the state-of-the-art.

- **Performance**: While still exploiting the structure information derived by LDA, we run PageRank only once and obtain similar or better keyphrases.
- **Configurability**: Users can balance topic specificity and corpus specificity of the extracted keyphrases and can tune the results according to use cases.

Conclusions & Possible Applications

- **Frontend features**
  - Headphone Buying Guide
  - Comparison Table

Find your perfect headphones

Based on customer reviews

Active Enthusiast

For those who like to break a sweat and are always on-the-go

CONTROL / DURABILITY / FIT

Casual Listener

For those who like to listen on long commutes and fill empty moments with sound

QUALITY / SOUND / PRICE

Music Lover

For those who are constantly pursuing the best listening experience

BALANCE / CLARITY / BASS

Jet Setter

For those who love to travel and are always dreaming of their next destination

NOISE CANCELLING / SIZE / COMFORT

●●○○○ Willkommen 14:09

Compare to similar items

	This item LG Electronics 32LF500B 720p L...	TCL 32S305 32-Inch 720p Roku S...	Sony KD 32-Inch
RATING	401 reviews	1125 reviews	537 revi
PRICE	From $199.39	$149.99	$298.00
SHIPPING	—	FREE Shipping	FREE Sh
SOLD BY	Available from these sellers	Amazon.com	Amazon
SATISFIED CUSTOMERS LIKED	picture quality (62) price (26) color (11) size (10)	picture quality (138) price (44) sound (41) quality (15)	picture price (1 quality size (5)

See more details

Have a question?

Find answers in product info, Q&As, reviews

Conclusions & Possible Applications

- **Frontend features**
  - Headphone Buying Guide
  - Comparison Table

- **Backend features**
  - Improving internal/external search results
  - Personalization
  - etc.

Salience Rank

Efficient Unsupervised Keyphrase Extraction with Topic Modeling

Weiwei Cheng

Amazon Development Center Germany

Berlin Germany

joint work with Nedelina Teneva