### Introduction

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.

### Overview

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. 1997), pairwise ranking (Jiang et al. 2006)
   - **Unsupervised**: graph-based ranking (Mihalcea & Tarau, 2004), topic-based clustering (Griffen et al., 2008), language modeling (Tumolo & Hunt, 2003)

### 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.

#### Definitions

- **The topic specificity of a word** \( w \): \( TS(w) = \sum_{t \in T} p(t | w) \log \frac{p(t | w)}{p(t)} \)
- **The corpus specificity of a word** \( w \): \( CS(w) = p(w | corpus) \)
- **The salience of a word** \( w \): \( S(w) = (1 - \alpha) CS(w) + \alpha TS(w) \), where \( \alpha \) is the tradeoff parameter balancing corpus and topic specificity of \( w \).

Our random walk:

\[
R(w) = \lambda \sum_{w_t \in w} \frac{e(w_t, w)}{Out(w)} R(w_t) + (1 - \lambda) S(w)
\]

Comparing to TPR, PageRank needs to be run only once.

### Experiments

- In terms of **performance**, while computationally more efficient, Salience Rank obtains comparable or better keyphrases on benchmark data.

### Conclusions

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.

#### Applications

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