Keyphrase Extraction Using Deep Recurrent Neural Networks on Twitter

Abstract: Keyphrases are important because they can provide a concise summary of the main ideas of a text. This can be helpful for users who want to quickly get the gist of a text or who want to find specific information in a large corpus of text.Traditionally, keyphrases have been extracted from documents or articles. However, due to the length limitations of Twitter, existing keyphrase extraction methods often perform poorly on tweets.The proposed deep recurrent neural network (RNN) model addresses this challenge by combining keyword information and context information to extract keyphrases from tweets. The RNN model is able to learn the relationships between words in a sequence, which is important for understanding the context of words in a tweet.
Keyphrases are important because they provide a concise and valuable way for users to grasp the main ideas within text.
Twitter Data Challenge: Extracting keyphrases from tweets is a unique challenge due to the character limitations on Twitter. These limitations often make existing keyphrase extraction methods less effective.
Proposed Approach: The paper introduces a novel deep recurrent neural network (RNN) model. This model is designed to combine both keywords and context information in the process of extracting keyphrases from tweets.
Large-Scale Dataset: To evaluate their proposed method, the authors collected a substantial dataset from Twitter. This dataset is used for experimentation and testing.
Positive Experimental Results: The research paper presents experimental results that demonstrate the superiority of their proposed method when compared to previous keyphrase extraction approaches. The method performs significantly better, showcasing its effectiveness in dealing with the unique challenges presented by Twitter data.
The research paper addresses a relevant and challenging problem of keyphrase extraction from Twitter data. It recognizes the limitations imposed by the brevity of tweets and proposes a deep learning solution to overcome this issue. The inclusion of a large-scale dataset from Twitter for evaluation adds credibility to the research.

1 Introduction
Traditional keyphrase extraction methods often perform poorly on tweets due to their short length. The proposed RNN model addresses this challenge by jointly modeling the keyword ranking, keyphrase generation, and keyphrase ranking steps.
Keyphrases Significance: Keyphrases are essential as they provide a concise way to capture the main topics within documents. Unlike keywords, keyphrases often consist of multiple words, making them more precise in meaning.
Methods for Keyphrase Extraction: The text mentions various methods for keyphrase extraction, including those using linguistic features, supervised classification, ranking-based approaches, and clustering-based techniques. These methods are typically applied to single or multiple documents.
Keyphrase Extraction from Tweets: Extracting keyphrases from tweets presents unique challenges due to Twitter's character limit. Some methods have been proposed for topic-based keyphrase extraction from tweet collections, considering the limited content length.
Proposed Approach: The research discussed in this text focuses on the challenging task of automatically extracting keyphrases from individual tweets. To address this, the authors propose a deep recurrent neural network (RNN) model. This model combines keyword ranking, keyphrase generation, and keyphrase ranking into a single process using two hidden layers.
Dataset Construction: To train and evaluate their method, the authors introduce a novel approach to construct a large dataset containing tweets with predefined keyphrases. This dataset construction method relies on hashtag definitions and their usage in tweets.
Contributions: The main contributions of this work include the introduction of the two-hidden-layer RNN model for joint keyphrase processing, the innovative dataset construction method, and experimental results demonstrating the superiority of their approach compared to existing methods.

2 Proposed Methods

The proposed model is a joint-layer recurrent neural network (joint-layer RNN) that jointly processes the keyword ranking, keyphrase generation, and keyphrase ranking steps. The model has two hidden layers and two output

layers. The first output layer predicts whether the current word is a keyword, and the second output layer predicts whether the current word is part of a keyphrase. The output layers are combined into a single objective function.
The model is trained using a dataset of tweets with golden standard keyphrases. The objective function is minimized using the stochastic gradient descent (SGD) algorithm.
The proposed model outperforms state-of-the-art methods on the task of keyphrase extraction from tweets.

2.1 Deep Recurrent Neural Networks:
This section introduces the concept of Deep RNNs, which are a way to capture contextual information within a sequence of words. Instead of simply concatenating neighboring features for input, a Deep RNN leverages an indefinite number of layers, introducing memory from previous time steps. The challenge with traditional RNNs is their lack of hierarchical processing. To overcome this, Deep Recurrent Neural Networks (DRNNs) are discussed.

2.2 Joint-layer Recurrent Neural Networks:
This section introduces the proposed Joint-layer Recurrent Neural Network (joint-layer RNN) as a variant of the stacked RNN with two hidden layers. The joint-layer RNN contains two output layers that are combined into an objective layer. The network aims to determine whether a word is a keyword or part of a keyphrase. The text describes the architecture of this model, which simultaneously models keyword identification and keyphrase extraction.

2.3 Training:
The training process for the neural network is explained. The parameters for the network are learned through back-propagation, optimizing the model using the stochastic gradient descent (SGD) algorithm. The objective function for training involves discriminating keywords and keyphrases at different levels, with a linear weighted factor (α) for balancing these objectives. The objective function is minimized to train the model.

3 Experiment
3.1 Data Construction:
The authors collected a large dataset of more than 41 million tweets from various users. To create an evaluation dataset, they identified hashtags that could serve as keyphrases for tweets. Only Latin-character tweets without URL links and non-conversational content (tweets that do not start with "@username") were considered. The dataset consisted of 110,000 tweets with hashtags that were suitable for keyphrase extraction.

3.2 Experiment Configurations:
The dataset was split into training (70%), development (10%), and testing (20%) sets. The evaluation metrics used were precision (P), recall (R), and F1-score (F1). Word embeddings were used as input for the neural network, and pre-trained vectors based on a Google News dataset were employed. Default parameters included a window size of 3, 300 neurons in the hidden layer, and an α value of 0.5.

3.3 Methods for Comparison:
Several methods were used for comparison, including Conditional Random Fields (CRF), Recurrent Neural Networks (RNN), Long Short-Term Memory (LSTM), Recurrent Conditional Random Fields (R-CRF), and Automatic Keyword Extraction on Twitter (AKET).

3.4 Experiment Results:
The table labeled "Table 2" in the text shows the performance of different methods for keyphrase extraction. The Joint-layer RNN method outperformed all other methods, with an F1-score of 80.97%, indicating its superiority.
To analyze the results of keyword extraction, the Joint-layer RNN method was compared to AKET. The table labeled "Table 3" in the text shows that while AKET had a slightly higher recall, the Joint-layer RNN method performed significantly better in precision and F1-score, indicating its better performance in keyword finding.
Further sensitivity analysis was conducted on hyperparameters. It was observed that the number of neurons in the hidden layers, window size, and the α value did not significantly affect the model's performance.

Additionally, the authors evaluated the impact of word embedding by comparing word embeddings with or without updates and random word embeddings with or without updates. Word embedding updates provided better performance.
The experiment also assessed the effect of training data size, indicating that the model's performance improved as more training data was used.
The text also includes figures showing the performance with different numbers of neurons in hidden layers, window sizes, α values, percentage of training data, and the convergence of the training process.

Review:
This section provides a comprehensive overview of the experiments conducted to evaluate the proposed keyphrase extraction model for Twitter data. The results demonstrate the effectiveness of the Joint-layer RNN method in comparison to other state-of-the-art methods. Sensitivity analyses and comparisons with AKET, a related method, further strengthen the findings.

Hyper-parameter analysis

- Number of neurons in the hidden layers
- Window size
- α value

4 Related Work

Supervised Machine Learning Approaches:
Supervised keyphrase extraction methods treat the problem as a classification task, where candidates are classified as either keyphrases or non-key phrases based on annotated training data. Notable contributions and methods in this category include:
KEA: Developed by Frank et al. (1999), KEA used features like term frequency-inverse document frequency (tf-idf) and the first occurrence of terms as input for Naive Bayes classification.

Hulth (2003): This approach used linguistic knowledge, including part-of-speech tags, to identify candidate phrases from text.

Tang et al. (2004): Applied Bayesian decision theory for keyword extraction.

KEA++: An extension of KEA developed by Medelyan and Witten (2006), which incorporated semantic information from domain-specific thesauri to enhance keyphrase extraction.

Unsupervised Ranking Approaches:

In unsupervised approaches, keyphrase extraction is treated as a ranking problem, and various metrics are used to rank terms or phrases in documents. The key methods in this category include:

TF-IDF: A classic approach based on term frequency-inverse document frequency (TF-IDF) that ranks terms based on their frequency and inverse document frequency.

TextRank: Proposed by Mihalcea and Tarau (2004), TextRank constructs keyphrases using PageRank values on a graph extracted from the text.

Clustering-Based Approaches: Liu et al. (2009) introduced a clustering-based approach to ensure that keyphrases extracted semantically cover the document.

Statistical Mechanics: Ali Mehri et al. (2011) introduced a method that ranks words in texts and classifies the correlation between word-type occurrences using non-extensive statistical mechanics.

5 Conclusion

- Keyphrase extraction is the task of identifying the most important words and phrases in a text. Keyphrases can be used for a variety of tasks, such as text summarization, classification, and information retrieval.
- Single tweets are short text messages that are posted on the social media platform Twitter. Tweets are limited to 280 characters, which makes them challenging to extract keyphrases from.
- Deep recurrent neural networks (RNNs) are a type of machine learning model that is well-suited for processing sequential data, such as text. RNNs can learn to identify patterns in text and make predictions about the next word or phrase in a sequence.

The proposed RNN model for keyphrase extraction works as follows:

1. The model takes a tweet as input and represents it as a sequence of vectors. Each vector represents a word in the tweet.
2. The model then passes the sequence of vectors through two hidden layers. The first hidden layer is responsible for ranking the keywords in the tweet. The second hidden layer is responsible for classifying the keywords as either keyphrases or non-keyphrases.
3. The outputs of the two hidden layers are combined into a final objective function. The objective function is trained to minimize the error between the predicted keyphrases and the ground truth keyphrases.

The proposed model was evaluated on a dataset of crawled tweets. The dataset was filtered to only include tweets that contained at least one ground truth keyphrase. The model achieved better results than the state-of-the-art methods on the dataset.

The proposed model has several advantages over previous methods for keyphrase extraction on single tweets:

- The model can jointly learn to rank keywords and generate keyphrases. This allows the model to learn the relationships between keywords and to identify keyphrases that are more informative than individual keywords.
- The model uses a deep recurrent neural network architecture. This allows the model to learn long-range dependencies in the text, which is important for keyphrase extraction.
- The model is trained on a large dataset of crawled tweets. This allows the model to learn a robust representation of the language and to extract keyphrases from a variety of different tweets.