Catchphrase Extraction from Legal Documents Using LSTM Networks

Rupal Bhargava[1] Sukrut Nigwekar[2] Yashvardhan Sharma[3]

WiSoc Lab, Department of Computer Science

Birla Institute of Technology and Science, Pilani Campus, Pilani-333031

{rupal.bhargava1, f20150292, yash3} @pilani.bits-pilani.ac.in

ABSTRACT

Legal texts usually have a complex structure and reading through them is a time-consuming and strenuous task. Hence it is essential to provide the legal practitioners a concise representation of the text. Catchphrases are those phrases which state the important issues present in the text, thus effectively characterizing it. This paper proposes an approach for the subtask 1 of the task IRLed (Information Retrieval from Legal Documents), FIRE 2017. The proposed algorithm uses a three step approach for extracting catchphrases from legal documents.

CCS Concepts

• Information systems ! Retrieval tasks and goals • Information systems ! Information extraction

Keywords

Keyword Extraction; Legal Documents; Deep Learning; LSTM; Natural Language Processing; Information Retrieval

1. INTRODUCTION

A prior case (also called a precedent) is an older court case related to the current case, which discusses similar issue(s) and which can be used as reference in the current case. If an ongoing case has any related/relevant legal issue(s) that has already been decided, then the court is expected to follow the interpretations made in the prior case. For this purpose, it is critical for legal practitioners to find and study previous court cases, so as to examine how the ongoing issues were interpreted in the older cases.

Generally, legal texts (e.g., court case descriptions) are long and have complex structures. This makes their thorough reading time-consuming and strenuous. So, it is essential for legal practitioners to have a concise representation of the core legal issues described in a legal text. One way to list the core legal issues is by keywords or key phrases, which are known as "catchphrases" in the legal domain.

In order to address this issue FIRE 2017 organized a task to extract catchphrases from legal documents. The task was to given training set of documents and their corresponding catchphrases, extract catchphrases from new documents.

Rest of the paper is organized as follows. Section 2 explains the related work that has been done in the past years. Section 3 describes the dataset provided by IRLed 2017 organizers. Section 4 explains the proposed technique that has been performed. Section 5 elaborates the evaluation and error analysis. Section 6 concludes the paper and presents future work.

2. RELATED WORK

Various techniques are being used for the task of keyword extraction [12]. They are broadly divided into supervised learning, unsupervised learning and heuristic based. The goal of supervised learning approaches was to train a classifier on documents annotated with keyphrases to determine whether a candidate phrase is a keyphrase (Witten et al., 1999; Frank et al., 1999) [4]. Another approach was to build a ranker for keyword ranking (Jiang et al., 2009) [11].

Unsupervised techniques proposed can be categorized into four groups. Graph-based ranking is based on the idea to build a graph from input document and rank its nodes according to their importance using a ranking method (e.g., Brin and Page (1998)) [10]. Topic-based clustering involves grouping the candidates into topics such that each topic is composed of only and only those candidates (Grineva et al., 2009) [5]. Simultaneous learning approach is based on the assumption that important words occur in important sentences and a sentences is important is it contains important words (Wan et al. (2007)) [9]. Language modeling scores keywords based on two features, namely, phraseness and informativeness (Tomokiyo and Hurst (2003)) [8].

Typical heuristics include (1) using a stop word list to remove stop words (Liu et al., 2009b) [7], (2) allowing words with certain part-of-speech tags (e.g., nouns, adjectives, verbs) to be candidate keywords (Mihalcea and Tarau, 2004) [6], (3) allowing n-grams that appear in Wikipedia article titles to be candidates (Grineva et al., 2009) [5], and (4) extracting n-grams (Witten et al., 1999) [4] or noun phrases (Barker and Cornacchia, 2000) [3] that satisfy pre-defined lexico-syntactic pattern(s) (Nguyen and Phan, 2009) [2].

3. DATASET DESCRIPTION

Dataset provided by the organizers [1] contained two sets of legal texts – training and testing. The training set was accompanied by the catchphrases corresponding to each text. The given catchphrases mainly consisted of words present in the text and rarely included phrases which were not present in the document.

4. PROPOSED TECHNIQUE

The problem is formulated as a classification task and the objective is to learn a classifier using LSTM network. The proposed methodology involves a pipelined approach and is divided into four phases:

- Pre-processing
- Candidate phrase generation
- Creating vector representations for the phrases
- Training a LSTM network

4.1 Pre-Processing

The legal texts were pre-processed in order to ensure uniformity. Pre-processing included removal of special characters, numbers and words which were not present in the English dictionary and converting all characters to lower case.

4.2 Candidate Phrase Generation

To generate candidates, n-grams with n in range 1 to 4 were created from the text. A standard stop list of common English words is taken to reduce the candidates. If the candidate starts or ends with a stop word then it is removed. To reduce candidates further an assumption was made that, words adjacent to given catchphrase will not be catchphrases. The assumption is justified as catchphrases are identified by removing stop words; conversely stop words can be generated by removing catchphrases. This modification to the stop list was done simultaneously with generating catchphrases. The method carries an inherent bias as the candidates generated from documents used in the beginning will be chosen according to a smaller stop list and those in the end will be according to a larger list. To remove this bias the documents were chosen randomly to generate candidates.

4.3 Creating Vector Representation

Word vector representations were created using Google News word-2-vec model. For phrases containing more than one word, word vectors were combined by obtaining their weighted average with the weights being the TFxIDF score of the constituent words.

4.4 Training the Model

Long-Short Term Memory units were used because text is considered to be a continuous input as the words used earlier can affect words used later in the text. Keras framework on top of TensorFlow backend was used to build the model. The number of LSTM units in the model was 100, dropout was set to 0.5 and a dense layer was added at the end to combine the outputs of the units to give a probability.

5. EVALUATION RESULTS

The proposed method achieved mean average precision of 0.0931 and overall recall of 0.0988. The precision could be probably improved by using a different model. Although the results are not very good this does not rule out the possibility of using deep learning for the task.

6. CONCLUSION

Catchphrases present a summary of a legal text and are very useful for practitioners. They can be used to implement a document retrieval system as they can be used as representation of the document needed. This working note presents an extraction system using LSTM network. The results are poor but LSTM are suited to the task at hand because of its continuous nature and hence should be explored further.

References

[1] Mandal, K. Ghosh, A. Bhattacharya, A. Pal and S. Ghosh. Overview of the FIRE 2017 track: Information Retrieval from Legal Documents (IRLeD). *In Working notes of FIRE 2017 – Forum for Information Retrieval Evaluation, Bangalore, India, December 8-10, 2017*, CEUR Workshop Proceedings. CEUR-WS.org, 2017.

[2] Chau Q. Nguyen and Tuoi T. Phan. 2009. An ontology-based approach for key phrase extraction. *In Proceedings of the Joint Conference of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing: Short Papers*, pages 181–184.

[3] Ken Barker and Nadia Cornacchia. 2000. Using noun phrase heads to extract document keyphrases. *In Proceedings of the 13th Biennial Conference of the Canadian Society on Computational Studies of Intelligence*, pages 40–52.

[4] Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. KEA: Practical automatic keyphrase extraction. *In Proceedings of the 4th ACM Conference on Digital Libraries*, pages 254–255.

[5] Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. *In Proceedings of the 18th International Conference on World Wide Web*, pages 661–670.

[6] Rada Mihalcea and Paul Tarau. 2004. TextRank: Bringing order into texts. *In Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 404–411.

[7] Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009b. Clustering to find exemplar terms for keyphrase extraction. *In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 257–266.

[8] Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. *In Proceedings of the ACL Workshop on Multiword Expressions*, pages 33–40.

[9] Xiaojun Wan, Jianwu Yang, and Jianguo Xiao. 2007. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. *In Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 552–559.

[10] Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual Web search engine. *Computer Networks*, 30(1–7):107–117.

[11] Xin Jiang, Yunhua Hu, and Hang Li. 2009. A ranking approach to keyphrase extraction. *In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 756–757.

[12] Kazi Saidul Hasan and Vincent Ng. 2014. Automatic Keyphrase Extraction: A Survey of the State of the Art. *In Proc. of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 1262-1273.