ReDefine - Research Definitions Summarized

Nikhita Sharma, Dig Vijay Kumar Yarlagadda, Harsha Komalla, Sai Madhavi Sunkari
University of Missouri - Kansas City

Abstract – *Automatic summarization is a complex task often requiring multiple steps to achieve a sensible summary. Research article summarization is long done by using extractive summarization techniques, which resulted in a summary that is obscure or which leaves out major definitions or parts of an article. In this paper, we introduce a system called ReDefine, which uses relation extraction to generate a summary in interactive graphical manner. ReDefine is unique in that it preserves all the major terms or definitions in an article along with their usage and provide readers with a clear overview of the summarized article.*

1. INTRODUCTION

The motivation behind ReDefine stemmed from having to read dozens of research articles to understand all the terms in a single research paper. ReDefine intend to provide a concise summary of the research articles for a reader who is completely unaware of content that is discussed in the article. This system has the potential to be useful in many different ways: a beginner in a field can explore research papers in a particular domain, a search engine can crawl through the graph for more relevant search results, etc.,

The objectives of our project are

- Building a scalable system for summarization of research articles which would give a graphical representation of the contents of the research paper including the important key terms and the relation between them. This would help understand the contents of the paper and the topic of discussion.

- Classify the research article into sub-categories based on terms used in the article.

We intend to develop ReDefine with following features:

- User can provide a corpus of research papers related to a field as input.

- Users are presented with an interactive graph, which can be explored to view the interactions between terms in research categories under the research field, authors, co-authors and interaction between related articles.

2. RELATED WORK

The research in automatic summarization in last decade focused majorly in two areas: extractive and abstractive summarization. The report, "Advances in Automatic Text Summarization", Inderjeet Mani et al.[1], details many such approaches. Moreover, Information retrieval techniques are used in system detailed in "Automatic text structuring and summarization", Gerard Salton et al.[2], but it is too complex to for real-time implementations. Most of these systems require large input datasets and used requires huge amount of manual training to generate a good summary.

3. PROPOSED SOLUTION:

We propose a scalable and efficient system called ReDefine, which uses a combination of OpenIE[3] and NLP techniques to extract summary from a paper's content. OpenIE is used to extract relation triples from the document representing subject, relation and object triples. The document is initially divided into sentences, sentences are divided into clauses and each clause is divided into triples. We have tried different OpenIE versions and extensions like Stanford CoreNLP OpenIE[20], Ollie, AllenAI OpenIE and Reverb; AllenAI OpenIE 4.1 gives the best results among them. This is because OpenIE 4.1 can extract N-ary relations: relations with more than one object, as shown in example below.

Normal relation: *We, describe, a set of typical MLbase use cases*.

N-ary relation: *We, describe, [a set of typical MLbase use cases, the optimizer's model selection, the MLbase run-time])*

This is particularly useful when we are trying to extract the usage of a single subject in more than one place in the paper, thereby abstracting the essence of its usage in the entire paper. Extractive summarization techniques often cannot achieve this; they can only decide whether to include an entire sentence containing a subject or not.

Further, the paper is classified into a pre-defined set of categories using an ontology-based approach to further help the reader of the paper understand the domain of the paper.

4. IMPLEMENTATION

GitHub repository:
https://github.com/nikhitasharma/KDM_Summer_2016_Cognitos

Video:
https://youtu.be/zCQ2xim8Wcc

4.1 Architecture:

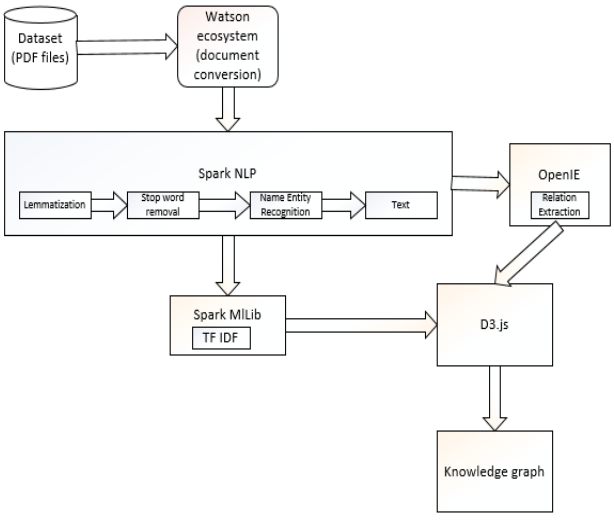

Fig 4.1: Architecture Diagram

The architecture of ReDefine system is shown in figure 4.1. The components in it are explained below.

Watson Document Conversion service:

Watson Ecosystem provides a Document Conversion service to convert PDF or HTML documents into plain text or JSON files. This service uses a proprietary Optical Character Recognition (OCR) technique and supports content in various languages.

Spark NLP:

ReDefine uses NLP tools provided by Spark MLLib[10]. Spark MLLib is chosen over Stanford CoreNLP, due to its scalability and compatibility with Apache Spark. We updated stop-word list, to a more comprehensive list provided by MySQL[4]. Lemmatization is done to obtain only bases of each word. We have implemented a pipeline to process all these NLP tasks in tandem across the entire input text corpus.

Though we do not directly use them in our summary, we tried to extract author names using Name Entity Recognition (NER). We observed that NER doesn't always deliver the consistent results. As an alternative, we suggest extracting metadata using Apache PDFBox[5].

Open IE:

Open IE 4.1 is a scala based system which uses ChunkedExtractor and Srlie to extract relations in a input document. Here is an example of relation extraction using Open IE 4.1[6]:

Sentence: The U.S. president Barack Obama gave his speech on Tuesday to thousands of people.
Extractions:
(Barack Obama, is the president of, the U.S.)
(Barack Obama, gave, his speech)
(Barack Obama, gave his speech, on Tuesday)
(Barack Obama, gave his speech, to thousands of people).

We modified the source code of Open IE to extract relations and filter them based on top 30 TFIDF words and confidence scores of extractions.

D3.js:

D3.js is a JavaScript library which provides dynamic visualizations using SVG, HTML5 and CSS standards. SVG-objects are created by pre-built JavaScript functions and styled using CSS. We used Radial Tidy Tree and Collapsible Tree[7] to represent our relations.

Ontology:

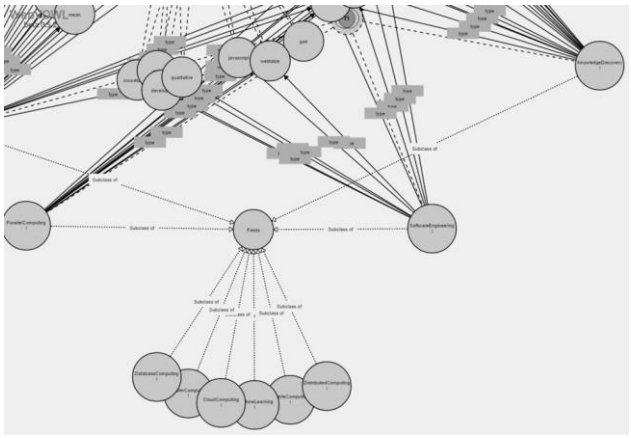

Fig 4.2: Ontology of ReDefine system

An ontology is dynamically created using Naive-Bayes and LDA for discovering topics among the text corpus, as shown in figure 4.2. The text corpus has 10 categories, each representing a sub-field of computer science (Cloud Computing, Cluster Computing, Big Data etc.,). A model is generated based on the text corpus. When a new document is input to the model, it can detect various topics in the document and it can also classify terms in it as one of the ten categories.

4.2 Workflow:

We initially prepare a dataset from input documents by converting PDF documents into text documents using Watson Document Conversion service. Fig 4.3 shows the workflow of our system. We process the text data using NLP tools, extract relations using Open IE, filter those relations

using TFIDF and represent them in a graphical format.

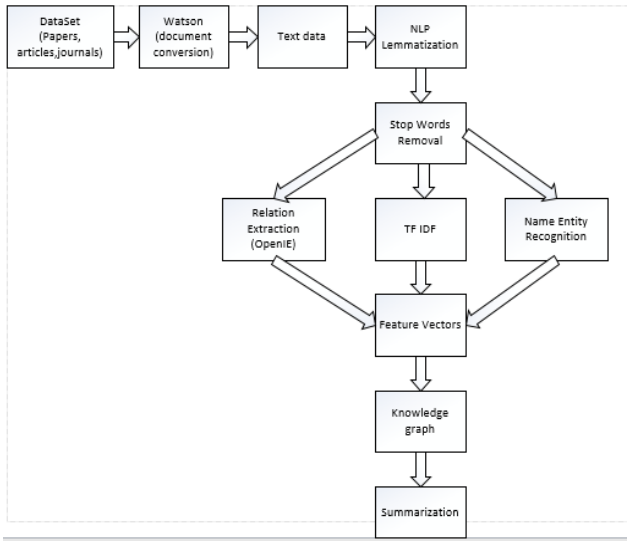

Fig 4.3: Workflow Diagram

5. RESULTS AND EVALUATION

The system is trained on 128 publications/articles which are categorized into 10 different categories, each field being a sub-field of computer science: Big Data, Cloud Computing, Cluster Computing, Database Computing, Distributed Computing, Knowledge Discovery, Machine Learning, Mobile Computing, Parallel Computing and Software Engineering. For a given input document, a graph summary is displayed. Figure 5.1 shows graph summary for the input document "MLbase: A Distributed Machine-learning System", Tim Kraska et al.[8]

Graph Summary will give user an overview of important terms in the document. User can expand each term to further understand how that term is used in the document.

The summary results are encouraging when compared to traditional abstractive or extractive summarization techniques, as discussed in "A Survey on Automatic Text Summarization", Dipanjan Das et al[9].

The system exhibited a precision and accuracy ranging from 0.8 - 0.9.

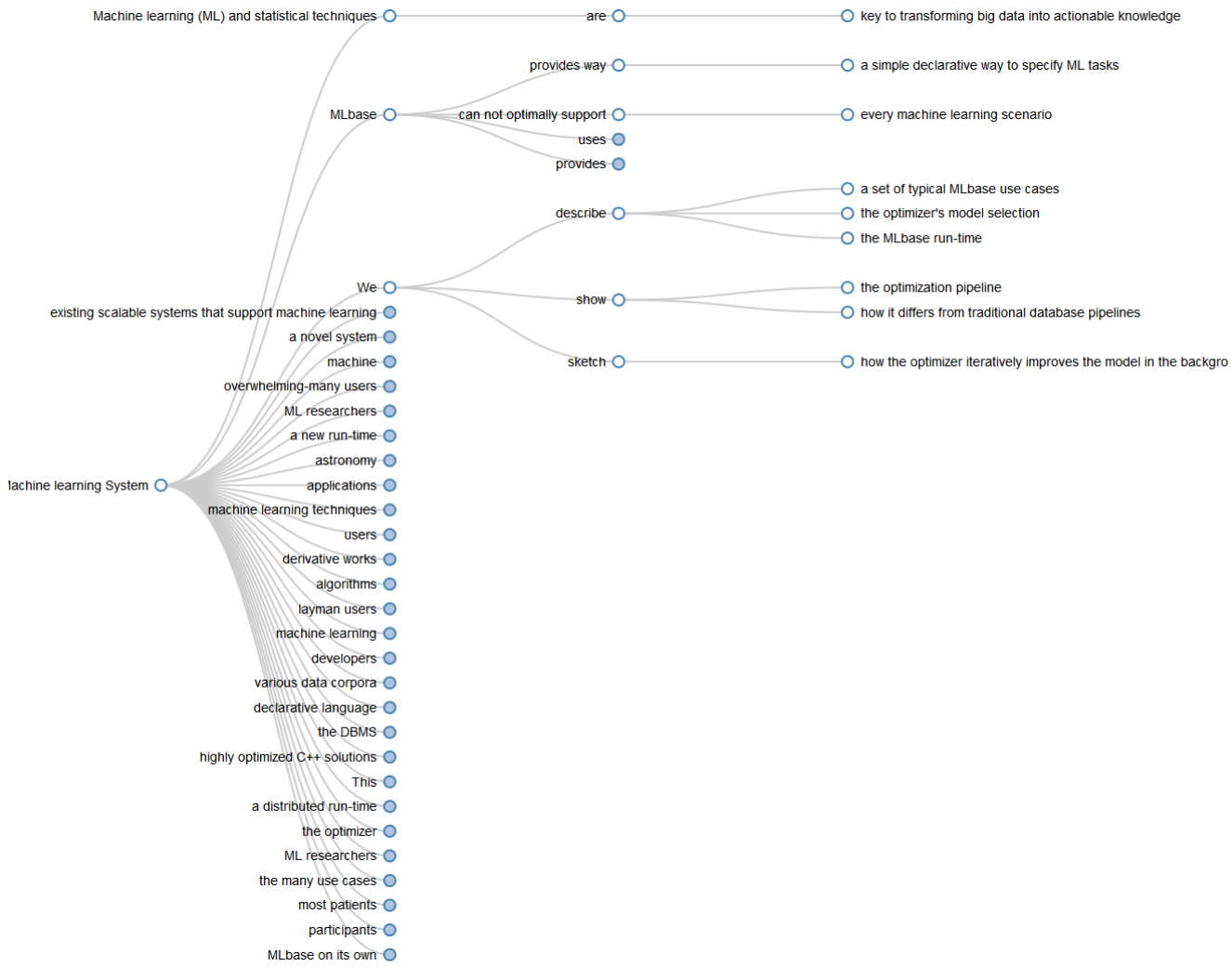

Fig 5.1: Graph Summary of research article

On a computer running on Intel i7-6700HQ CPU and 12 GB RAM assigned to the ReDefine system, the system took 12 minutes to train the model on the input dataset of 128 articles and an average of 1.907 seconds to train the classify terms in document. For relation extraction, initial loading of libraries for Open IE 4.1 took 6 minutes, and an average of 2.03 seconds to extract relations from a 2300-word input document. Thus the system exhibited a suitability for real-time systems, where the summary can be generated in less than 5 seconds. The summarization rate is about 10% of the input document.

6. CONCLUSION

ReDefine is summarization tool uses relation extraction to summarize an input document. It is a unique take on traditional automatic summarization techniques, which are too complex and majorly focuses on summarization rate rather than clarity of summary. The system is simple, efficient, scalable and can work as a real-time system.

7. FUTURE WORK

The ReDefine system is scalable, but its performance is never calculated for large datasets, due to lack of

availability of such datasets for articles/publications. Moreover, in addition to a graph summary, we plan to extend this system to include a text summary.

ACKNOWLEDGEMENTS:

We'd like to thank the ideas, guidelines and suggestions given by Dr. Yugyung Lee, Mayanka Chandrashekar and Vadlamudi Naga Krishna. We'd also like to authors of OpenIE upon which ReDefine is developed.

REFERENCES

[1]. Mani, Inderjeet, and Mark T. Maybury, eds. *Advances in automatic text summarization*. Vol. 293. Cambridge, MA: MIT press, 1999.

[2]. Salton, Gerard, et al. "Automatic text structuring and summarization."*Information Processing & Management* 33.2 (1997): 193-207.

[3]. Fader, Anthony, Stephen Soderland, and Oren Etzioni. "Identifying relations for open information extraction." Proceedings of the Conference on Empirical Methods in Natural Language Processing. Association for Computational Linguistics, 2011.

[4].http://dev.mysql.com/doc/refman/5.7/en/fulltext-stopwords.html

[5]. https://pdfbox.apache.org/

[6]. https://github.com/allenai/openie-standalone

[7]. https://bl.ocks.org/mbostock/4339083

[8]. Kraska, Tim, et al. "MLbase: A Distributed Machine-learning System." CIDR. Vol. 1. 2013.

[9]. Das, Dipanjan, and André FT Martins. "A survey on automatic text summarization." *Literature Survey for the Language and Statistics II course at CMU* 4 (2007): 192-195.

[10]. Meng, Xiangrui, et al. "Mllib: Machine learning in apache spark." JMLR 17.34 (2016): 1-7.

[11].http://spark.apache.org/docs/latest/mllib-guide.html

[12].https://spark.apache.org/docs/1.1.0/mllib-feature-extraction.html

[13]. https://github.com/watson-developer-cloud

[14]. https://github.com/stanfordnlp/CoreNLP

[15]. http://nlp.stanford.edu/software/openie.html

[16].http://openie.allenai.org/

[17].http://reverb.cs.washington.edu/

[18].http://knowitall.github.io/ollie/

[19]. Open Information Extraction from the Web *Michele Banko, Michael J Cafarella, Stephen Soderland, Matt Broadhead and Oren Etzioni, Department of Computer Science and Engineering University of Washington.*

[20]. Manning, Christopher D., et al. "The Stanford CoreNLP Natural Language Processing Toolkit." ACL (System Demonstrations). 2014.