2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)

Keyword Extraction Performance Analysis

Abhishek Kumbhar
University of Michigan - Dearborn
Dearborn, USA
akumbhar@umich.edu

Mayuresh Savargaonkar
University of Michigan - Dearborn
Dearborn, USA
mayuresh@umich.edu

Aayush Nalwaya
University of Michigan - Dearborn
Dearborn, USA
analwaya@umich.edu

Chengqi Bian
University of Michigan - Dearborn
Dearborn, USA
bianc@umich.edu

Mohamed Abouelenien
University of Michigan - Dearborn
Dearborn, USA
zmohamed@umich.edu

***Abstract*— This paper presents a survey-cum-evaluation of methods for the comprehensive comparison of the task of keyword extraction using datasets of various sizes, forms, and genre. We use four different datasets which includes Amazon product data - Automotive, SemEval 2010, TMDB and Stack Exchange. Moreover, a subset of 100 Amazon product reviews is annotated and utilized for evaluation in this paper, to our knowledge, for the first time. Datasets are evaluated by five Natural Language Processing approaches (3 unsupervised and 2 supervised), which include TF-IDF, RAKE, TextRank, LDA and Shallow Neural Network. We use a ten-fold cross-validation scheme and evaluate the performance of the aforementioned approaches using recall, precision and F-score. Our analysis and results provide guidelines on the proper approaches to use for different types of datasets. Furthermore, our results indicate that certain approaches achieve improved performance with certain datasets due to inherent characteristics of the data.**

Keywords- NLP; Keyword Extraction; Text Mining;

I. INTRODUCTION

Keywords are the summative explanation of any longer text or a document in one word or a small phrase (often called a key phrase). Keyword extraction task is very common in various applications ranging from scientific publications to mining people’s opinion on products and services on online shopping portals, posts, reviews and question-answering systems, etc. Identifying and extracting relevant features for gathering the semantic meaning of a document by a keyword is a challenging task, especially on smaller datasets or metadata. We limit our scope to extracting ‘unigrams’ for this study.

Multiple approaches have been proposed for keyword extraction for tasks, such as opinion mining [1], text summarization [2], and text categorization [3]. Researchers have previously conducted studies to compare performance of various NLP methods on different datasets. In their comparison, they reported that RAKE superseded TextRank using text from journal papers in computer science [4] and LDA outperformed TF-IDF using environmental data with the presence of sufficient textual data [5].

To evaluate the performance of different models on varying types and sizes of datasets, we examine five approaches, namely, TF-IDF, TextRank, RAKE, LDA and Deep Learning on four different datasets. In this paper, we aim to contribute to the scientific community by providing guidelines on baseline method(s) that can be used for different types of data based on different nuances in the tested datasets to aid future research. Often, it might be the case that the results are biased towards a certain method depending on the type of data of interest. This will provide a benchmark for the scientific community to evaluate the performance of any new or improved method(s) for a fair and true comparison. In a pursuit to achieve this, we also annotate an amazon product reviews dataset, which opens a new avenue for testing traditional methods on review-based datasets without having to look in complexities of semantic evaluations achieved by shallow neural network. Another aspect of the proposed comparative study lies in the implementation and comparison of the shallow neural network semantic evaluation in comparison with other supervised and unsupervised models.

II. METHODOLOGY

Our models are based on the bag-of-words (BOW) approach, which is a representation of a document by the frequency of occurrence of its words. The model does not consider grammar or the order of the words, which is a drawback for semantic evaluation. To stamp out this limitation, we implement word embedding Continuous Bag Of Words (CBOW) for the shallow neural network model. We evaluate five methodologies, which fall into two categories; Supervised Models and Unsupervised Models. Models in which we do not have a training dataset are known as Unsupervised Models [6]. This category includes TF-IDF, TextRank and RAKE. Models in which we train our models to learn from the given data and be able to make predictions for unseen instances are known as Supervised Models [7]. This category includes LDA and shallow neural network.

978-1-7281-1198-8/19/$31.00 ©2019 IEEE
DOI 10.1109/MIPR.2019.00111

A. *Unsupervised Methods*

Our first approach is Term Frequency-Inverse Document Frequency (TF-IDF). It is an unsupervised numerical statistic methodology that measures the importance of different words by finding the words that have the highest ratio of occurrence in a given document to the frequency of occurrence in the whole set of documents [8]. Term Frequency (TF) calculates the frequency of appearance of a word in a document and Inverse Document Frequency (IDF) measures the importance of a term. It presents results solely on higher occurrence of words in a document, regardless of the semantic meaning of the text. In [9], usage of a preprocessing function known as Average Occurrence Frequency (AOF) showed some promising results when used with TF-IDF on a twitter-based dataset. AOF calculates the overall frequency of unique words in proportion to the total number of unique words in the data. This is of particular interest to us given the size of the datasets. Accordingly, we implement TF-IDF with and without AOF.

Our second approach is Text-Rank, which is a graph-based, language independent unsupervised ranking algorithm that determines keywords based on the importance of a vertex within a word graph, which is created by the BOW model for each document. Tokens are extracted from each sentence and part of speech tagger assigns a tag such as noun, verb or adjective to each token. POS tagging is integrated in the data to enable the syntactic filtering. Word stemming is not applied to avoid disputation with POS tagging. A relationship is created between the words in the graph and the documents. In addition, multiple iterations are implemented until a stabilized word score is achieved. Accordingly, top words with the highest scores are extracted as keywords. Usually 20-30 iterations are implemented until a convergence of 0.0001 is achieved [10].

Our third approach is Rapid Automatic Keyword Extraction (RAKE). RAKE is a domain and language-independent, unsupervised method for extracting keywords from individual documents. It is based on the individual frequency of words and its co-occurrences with other words in a document [4]. The selected words that occur are accumulated in a word graph called the co-occurrence graph. The co-occurrence graph of the words and the phrases are built to identify the frequency of association, after splitting the document into an array of words by specified delimiters. Using the frequency, we calculate the individual word score as the degree (number of times it appears + number of additional words it appears with) of a word divided by its total frequency. The result is a weight that favors longer phrases. A score is then calculated for each phrase, which is the sum of the individual word scores from the co-occurrence graph.

B. *Supervised Methods*

Our fourth approach is Latent Dirichlet Allocation (LDA). LDA is a generative statistical model that allows sets of observations, or words, to be sorted in groups based on topics. Each document is assumed to represent multiple topics and different words can represent topics with different probabilities. LDA takes words from all the documents assuming they are related across and constructs its model. The model further considers hyperparameters; alpha (per document topic distribution) and beta (per topic word distribution) to generate a matrix consisting of topics (rows) formed by words (columns) with their respective probabilities indicating the chance that they belong to that topic [11]. The underlying assumption is that the same topics would be found in training and test datasets. LDA assigns topics (keywords) from each document using a probability distribution and iterates till it achieves the maximum probability for the number of keywords we require.

Our last approach includes using a shallow network. Using this approach, a document is represented using a vector space model based on a BOW encoding of terms in a document. For our shallow neural network evaluation, we represent a document as a fixed length string vector with its elements being the frequency of occurrence of the words in a document. A major weakness of BOW encoding is that local word order is lost, and models tend to suffer high dimensionality and sparsity, requiring special memory efficient encoding schemes. Word2vec [12] is a recent state-of-the-art document representation model that uses a shallow neural network encoding to generate word vectors in a vector space, where vectors that are close to each other are semantically related. For our shallow neural network method, word2vec is used to extract the text features from the training dataset. There are two types of Word2vec models: CBOW and Skip-Gram [13]. Both require large dataset training to generate vectors as the word representation. In our experiment, we use word2vec libraries from Gensim [14]. After training the model, keywords are extracted using TextRank & Word2vec model. We do not use a validation set to tune the parameters during the Word2vec training, instead, default setting is adopted as is suggested in Gensim.

III. EXPERIMENTAL SETTINGS

In this section, we discuss our experimental setup for the approaches using four different datasets. We used ten- fold cross-validation scheme and evaluated our results using recall, precision and F-score. Words are stemmed using porter stemmer [15] for a fair comparison of the results.

A. *Preprocessing*

Initially, we tokenize our datasets i.e. to break the document into single words- ‘tokens’, for further processing. Stop words are the most common words used in any language, such as “she”, “him”, “it”, etc., which are irrelevant to the significance of our keyword extraction task. We use the Default English stop word list from [16] to remove these words. Hence, we get rid of them as well. Brackets, hyphens, and other such characters are removed to further clean the data for our analysis. All words are converted to lowercase to avoid errors. We also modify TF-IDF by using AOF as discussed earlier. Note that none of the other methods employ this approach.

B. Datasets : Training and Testing

For the purpose of our study, we use 4 datasets with varying forms, sizes and types with the aim of recommending benchmark methods to analyze each type of dataset.

1) Amazon

This is a dataset of user reviews for products sold online at Amazon[1]. We have selected first 100 reviews from Automotive category under Amazon Product data. These reviews do not have a specified structure, contain grammatical and spell errors, use slangs and have an average of 50 words in each review. To the best of our knowledge, this dataset has not been annotated before and hence we asked two annotators to manually extract keywords from the reviews. The inter-rater agreement is measured using Cohen's Kappa. It takes into consideration the possibility of an agreement between the annotators by chance and is calculated to be 0.728.

2) SemEval 2010

SemEval2010[2] dataset consists of 280 formal scientific articles, which were collected from the ACM Digital Library This dataset is well structured with specified heading and body sections. We use first 100 articles.

3) TMDB

Subsets of the TMDB[3] data are available for access to customers for personal and non-commercial use online. A subset of 1000 random data points (Sci-Fi Genre) were selected for this study. This data represents a brief abstract of movies.

4) Stack Exchnage

Stack Exchange[4] is a congregation of question-answer groups on varied topics and fields with each group covering a specific topic. Each answer to a question is subject to a review and award process. We randomly selected 1000 documents for this study.

C. Evaluation Metrics

Three evaluation metrics viz. precision, recall and F-measure (F1) are employed for evaluating the performance of methods [17]. We have used Recall-Oriented Understudy for Gisting Evaluation (ROUGE 2.0) method [18].

IV. Result and Discussion

In this section, we discuss and analyze our experimental results. Table 1 shows the performance using the five approaches measured against our four datasets using the three-evaluation metrics.

A. Evaluation Methods and Datasets

The LDA model works best for all three metrics on SemEval data as there is a probable concentration of topics as the whole corpus is a set of scientific documents. Furthermore, we suspect that the size of the dataset enhances the training process and hence improves the performance of LDA.

LDA works best for Amazon dataset as well due to the possible advantage of topic concentration given that the reviews are focused on automotive products with words such as, "service", "value" and "money" appearing more frequently than others. The availability of a training set seems to have an impact on identifying keywords from reviews having fewer words.

The TF-IDF model, as expected, works very well on Stack Exchange data since multiple users can answer the same questions. An important possible distinction between TF-IDF and TF-IDF (AOF) here is that AOF fails to retain, possibly important, 'keywords' below the average frequency. Compared to the two previous datasets, the performance, measured by all 3 metrics, drops for this dataset. A possible explanation is the presence of fewer words per document complimented by more documents, which complicates the keyword extraction process.

TF-IDF, LDA and RAKE achieve close performance for TMDB dataset as the movie's summaries are concentrated from a selected genre. Thus, we find similar topics and co-occurring keywords across the dataset which result in better performance by LDA, TF-IDF and RAKE respectively. In accordance with the previous observations, as the number of words per document increases, a slight increase can notice in all three metrics.

B. Recommendations and Guidelines

Based on our results, we observe that while selecting a baseline method for evaluation of any new method, it is important to consider some factors such as, the length of the document and number of documents in a corpus. We suggest following baseline methods based on the form, type and size of datasets.

- For datasets having scientific documents, such as SemEval, with a specified structure, correct grammar and approximately 5000 words, we recommend using LDA due to possible topic concentration.
- For datasets having reviews similar to amazon automotive product reviews with no structure, possible grammatical errors, slangs, and a small number of words, we recommend LDA.
- For datasets containing documents with multiple answers for same questions, such as the Stack Exchange data having correct grammar and 150-200 words, we suggest using TF-IDF.
- For datasets having documents with focused topic summaries, such as the TMDB data, correct grammar and a small number of words we suggest using LDA, TF-IDF(AOF) and RAKE in order of decreasing importance.

Conclusively, shallow neural network fails to compare with traditional methods due to the size of the training set. Hence, it will not prove a fair baseline for comparison for small datasets. A possible use of Word2vec along with Doc2vec and LDA might improve the results, which we leave for future work.

[1] http://jmcauley.ucsd.edu/data/amazon/

[2] https://github.com/boudinfl/semeval-2010-pre

[3] https://www.kaggle.com/tmdb/tmdb-movie metadata#tmdb_5000_movies.csv

[4] https://www.kaggle.com/c/transfer-learning-on-stack-exchange-tags

TABLE 1: RESULT OF METHOD'S PERFORMANCE BASED ON RECALL (R), PRECISION (P) AND F MEASURE (F1)

Datasets	**SemEval**			**Amazon Reviews**			**Stack Exchange**			**TMDB**		
	R	**P**	**F1**	**R**	**P**	**F1**	**R**	**P**	**F1**	**R**	**P**	**F1**
TF-IDF (AoF)	37.7	37.9	37.8	32.2	37.1	34.4	19.5	21.6	20.4	22.3	**24.0**	23.1
TF-IDF	35.0	35.4	35.2	28.5	30.4	29.4	**25.3**	**27.4**	**26.3**	**22.4**	**24.0**	**23.2**
RAKE	16.7	18.1	17.4	23.8	35.2	26.7	14.7	16.9	15.7	21.1	23.1	22.1
LDA	**41.8**	**43.4**	**42.6**	**33.6**	**38.8**	**36.0**	20.5	22.7	21.6	21.5	23.2	22.3
Text Rank	20.0	21.2	20.6	27.6	30.5	29.0	14.2	15.5	14.8	16.5	17.3	16.9
Shallow Neural Network	11.0	11.6	11.3	13.7	15.5	14.4	10.9	11.3	11.1	13.4	13.8	13.5

REFERENCES

[1] G. Berend, “Opinion expression mining by exploiting keyphrase extraction”, Proceedings of the 5th International Joint Conference on Natural Language Processing, Thailand 2011. pp 1162–1170

[2] Y. Zhang, N. Zincir-Heywood, and E. Milios, “World wide web site summarization”, Web Intelligence and Agent Systems: An International Journal 2.1, 2004. pp.39-53.

[3] A. Hulth and B.B. Megyesi, “A study on automatically extracted keywords in text categorization”, In Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics, Sydney 2006. pp. 537-544.

[4] S. Rose, D. Engel, N. Cramer and W. Cowley, “Automatic keyword extraction from individual documents”, Text Mining: Applications and Theory, 2010. pp.1-20.

[5] S. Tuarob, L.C. Pouchard and C.L. Giles, “Automatic tag recommendation for metadata annotation using probabilistic topic modeling”, In Proceedings of the 13th ACM/IEEE-CS joint conference on Digital libraries, USA 2013. pp. 239-248.

[6] Z. Ghahramani, "Unsupervised learning", In Advanced lectures on machine learning, Springer, Berlin, Heidelberg, 2004. pp. 72-112.

[7] S. B. Kotsiantis, I. Zaharakis & P. Pintelas, “Supervised machine learning: A review of classification techniques”, Emerging artificial intelligence applications in computer engineering, 2007. 160, 3-24.

[8] K. Sparck Jones, “A statistical interpretation of term specificity and its application in retrieval”, Journal of documentation 28.1, 1972. pp.11-21.

[9] S.K. Biswas, M. Bordoloi and J. Shreya, “A graph-based keyword extraction model using collective node weight”, Expert Systems with Applications 97, 2018. pp.51-59.

[10] R. Mihalcea and P. Tarau, “Textrank: Bringing order into text”, In Proceedings of the 2004 conference on empirical methods in natural language processing, 2004.

[11] X. Wan and T. Wang, “Automatic labeling of topic models using text summaries” In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, Germany 2016. pp. 2297-2305.

[12] Q. Le and T. Mikolov, “Distributed representations of sentences and documents”, In International Conference on Machine Learning, China 2014. pp. 1188-1196

[13] T. Mikolov, Q.V. Le and I. Sutskever, “Exploiting similarities among languages for machine translation”, arXiv preprint arXiv:1309.4168, 2013.

[14] R. Řehůřek and P. Sojka, “Gensim—Statistical Semantics in Python”, Statistical Semantics, Paris 2011.

[15] P. Willett, “The Porter stemming algorithm: then and now”, Program: electronic library and information systems 40 (3), 2006. pp. 219-223.

[16] https://www.ranks.nl/stopwords

[17] G. Hripcsak and A.S. Rothschild, “Agreement, the f-measure, and reliability in information retrieval”, Journal of the American Medical Informatics Association 12.3, 2005. pp.296-298.

[18] K. Ganesan, “ROUGE 2.0: Updated and Improved Measures for Evaluation of Summarization Tasks”, arXiv preprint arXiv:1803.01937, 2018