Keyword Extraction by Deep Learning
Mid Term Report

Yi Cheng(yicheng1), Anoop Hallur(ahallur), Xiaoqiu Huang(xiaoqiuh)

November 3, 2014

1 Introduction

As the goal of our project, we plan to do keyword extraction using deep learning. We are aiming to improve the performance of keyword extraction by deep learning techniques as compared to other techniques used previosuly. We believe that Deep learning techniques can improve the performance because when it has been applied to other similar tasks, a significant improvement has been observed. For example , Google speech recognizer uses Deep Neural Networks and accuracy is extremely high[cite].The area of applying deep learning to keyword extraction has not been explored much, hence we are trying to apply it and see how it performs.

2 Related Work

Before diving into the details of our model and algorithm, we want to summarize the related work that has been done in the field of Keyword Extractions, how they can be applied to our project and how deep learning can be applied to solve this particular problem and our thoughts on why deep learning should give better performance compared to other algorithms.

2.1 Baseline Algorithms of Keyword extraction

A survey was done by Lott.B[cite], and he summarizes that whenever he have a large corpus of data already available(as in our case), TF-IDF is the most accurate algorithms of the existing ones.

In TF-IDF model, we assign weight to each term in the model, and we choose the top'n' weighted words as the keywords for the text. The weight of each word is computed by taking into account the Term Frequency(TF) and Inverse Domain Frequency(IDF), where Term Frequency indicates how significant is the term to a spcicif document, and IDF takes into account how common the word is in the domain. We have implemented the benchmarks for this algorithms on our dataset.

Other algorithms presently being used are some variant of TF-IDF with domain specific modifications. For example , one technique uses a Bayes classifier with TF-IDF to compute the weights and extract the keywords. There are techniques which are to be used when we dont have a corpus of text available to us. Frequency based single Document Keyword Extraction is one such technique. This techniques coputes word weights by measuring the frequency of text occurence within two punctuation marks in the text. Since we have text corpus available, we did not want to be compared against these classes of algorithms.

2.2 Deep Learning Intution

3 Model Description

4 Experiment and Results

4.1 Baseline Algorithm Results

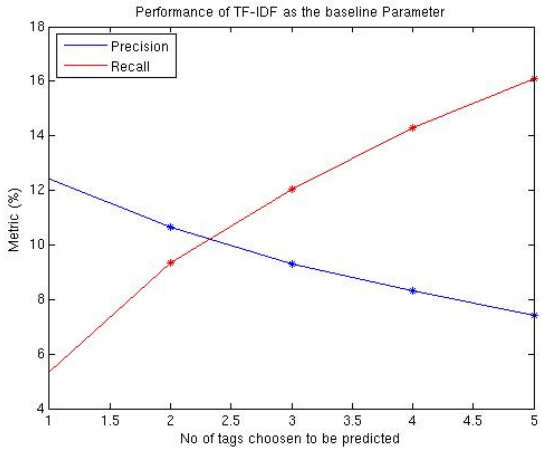
To evaluate the performance of our approach compared to existing techniques, we have used Precision and Recall rates as our benchmarking standards. We have formally defined these parameters as

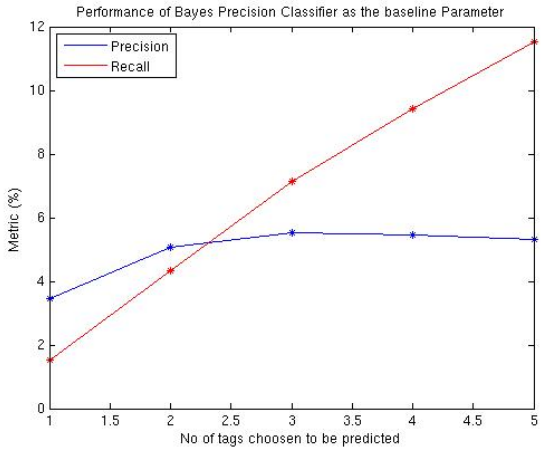
$Precision = \frac{sizeof(A \bigcap B)}{sizeof(A)}$

$Recall = \frac{sizeof(A \bigcap B)}{sizeof(A)}$

where A is the set of predicted tags and B is the set of actual tags.

We have implemnted TF-IDF and Bayseian Classifier approach as described by the survey for Keyword Extraction on our data set. With plain vanilla TF-IDF, the average precision and recall rates were less than 10 %. However, after preprocessing the data by using a stemmer, the accuracy is a modest 15-20% on topics in our data set for as the best performer. We have used portland stemmer, available as part of open source NLTK(Natural Language Processing Toolkit) project. The Bayesian Classifier is similar to TF-IDF apporach except that it takes into account the position of the word in text. The weight of each term is reduced as the log of the its position from the begining of the sentence.





As part of survey paper[6], we found out that the existing standard for key word extraction uses Lexical Tree approach and we are coming up with the precision and recall rates for Lexical Trees approach
{Report exact figures here}.

4.2 Dataset

We had identified data from stackexchange(`https://archive.org/download/stackexchange/stackexchange_archive.torrent`) to be a suitable data set for our project.

Of the 20 GB, available to us, we are using only 500 MB of data, on about 20 topics for our evaluation. We felt this was enough for testing purposes, and once we decide on the model, we will make use of all the data set available to us. Working with bigger datasets is a hassle when no concrete model is available and hence we stopped at 500MB. All our results and observations have been made on this 500MB of dataset, which we have cleaned and formatted for our applications.

5 Pending Work

6 References

1. Bengio Y, Schwenk H, Sencal J S, et al. Neural probabilistic language models[M]//Innovations in Machine Learning. Springer Berlin Heidelberg, 2006: 137-186.

2. Socher R, Perelygin A, Wu J Y, et al. Recursive deep models for semantic compositionality over a sentiment treebank[C]//Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP). 2013: 1631-1642.

3. Socher R, Lin C C, Manning C, et al. Parsing natural scenes and natural language with recursive neural networks[C]//Proceedings of the 28th International Conference on Machine Learning (ICML-11). 2011: 129-136.

4. Collobert R, Weston J, Bottou L, et al. Natural language processing (almost) from scratch[J]. The Journal of Machine Learning Research, 2011, 12: 2493-2537.

5. Matsuo Y, Ishizuka M. Keyword extraction from a single document using word co-occurrence statistical information[J]. International Journal on Artificial Intelligence Tools, 2004, 13(01): 157-169.

6. Lott B. Survey of Keyword Extraction Techniques[J]. UNM Education, 2012.

7. Hasan K S, Ng V. Automatic Keyphrase Extraction: A Survey of the State of the Art[J].