Project proposal: Key Words Extraction

Albina Akhmetgareeva, Aleksandr Safin, Ivan Rodin, Maxim Kaledin

November 21, 2017

Introduction

Keyword extraction is needed for automatic identification of terms which describe the subject of a document. By keywords we understand not only words, but also bigrams and key phrases.

Keyword extraction is a basic task in computational linguistics — readers and search engines benefit from keywords because they can decide more quickly whether the text is worth reading/showing. Website creators benefit from keywords because they can group similar content by its topics. Text processing algorithms benefit from keywords because they reduce the dimensionality of text to the most important features. And these are just some examples.

By definition, keywords describe the main topics expressed in a document. So, our task will be finding keywords for each text from some given text collection.

1 Problem Formulation

1.1 Extracting Keywords using Optimization of Modularity (M.Kaledin)

It is another approach using graphs of words. Suppose we have a matrix P of pairwise frequencies for each pair of words: p_{ij} stands for probability that words i and j are in the same sentence, according to the input data. This matrix can be considered as the adjacency matrix of some graph $G = (V, E)$, where V denote the words and E are the links weighted with p_{ij}. We could make a heuristic assumption, that clusters correspond to different big topics of the text. So, the extraction of keywords may be performed by clustering (or, community detection) and choosing the most frequent words from each big cluster.

The first algorithm suggesting to use modularity was Girvan-Newman algorithm [1]. It suggests to optimize the *modularity* Q of the given graph $G = (V, E)$, $|V| = n$, $|E| = m$. Suppose, we search the best graph partition into two communities (subgraphs with large density inside and less outside edges), let $s_i = 1$ if vertex i belongs to cluster 1 and -1 otherwise. The article suggests to consider the difference between the actual number of edges between two vertices and the expected one if they were randomly generated with uniform distribution. Thus

$$Q = \sum_{u,v \in V:\ u \neq v} \left(A_{ij} - \frac{k_u k_v}{2m} \right) (s_i s_j + 1)$$

defines the modularity of partition. It was noted that this approach can be generalized on case of several communities [2] with

$$Q = \sum_{u,v \in V:\ u \neq v} \left(A_{ij} - \frac{k_u k_v}{2m} \right) \delta(c_u, c_v),$$

where c_u, c_v denote community assignments and

$$\delta(c_u, c_v) = \begin{cases} 0 & \text{if } c_u = c_v \\ 1 & \text{otherwise} \end{cases}.$$

Also it is possible to use the hierarchical approach with 2-community clustering [3]. Usually optimization of modularity can be performed heuristically (for instance, the Louvain method [4]) and with spectral methods [3].

My goals are to suggest discrete optimization techniques for dealing with modularity optimization and to compare it with other methods which are considered in this proposal.

1.2 Matrix factorization-based approach (A. Safin)

It seems to be quite intuitive that keywords of the document reflect essential part of the whole document. So, one could think about the task of keywords extraction from the documents as the task of finding low-rank approximation of the term-document matrix, which intuitively contains keywords. Therefore, to extract keywords for each document simply means to choose k (for instance) words (or bigrams, if they are used it the bag of words vector) corresponding to the highest values.

So, the task could be formally described as:

$$\underset{\text{rank}A_r=r}{\text{minimize}} \|A - A_r\|_F^2,$$

where the r is the parameter of the model which could be optimized to provide acceptable results for key word extraction.

One approach to obtain such approximation is to use SVD and then truncate it up to r largest singular values [5].

Another approach utilize the following idea, that term-document matrix (probabilities of each word to be in the particular document) could be represented is a product of two matrices such that $p(w_i|d_j) = \sum_{k=1} p(w_i|t_k)p(t_k|d_j)$. In other words, one could consider the initial term-document matrix A as $A = WH$, where W is matrix determined by $p(w_i|t_k)$ and $H_{kj} = p(t_k|d_j)$. So, the most weighted elements in the j^{th} column of the matrix W reflects the keywords for the j^{th} topic. And matrix H reflects the probability of each topic for every document. Because we want consider the elements of H and W matrices as a probabilities (or proxy values for probabilities), it is natural to require that W and H would be non-negative.

The task of Non-negative Matrix Factorization [6] (NMF) could be formalized as:

$$\underset{W\geq 0, H\geq 0}{\text{minimize}} \|A - WH\|_F^2$$

Also, it is worth emphasizing that in [7] proposed the usage of SVD for NMF problem.

Summing up, in this direction of the project, we would try to use above mentioned technique and realize what results could be achieved by using them. Also, because the term-document matrix usually has sparse structure, therefore maybe there is a need in using some methods which proven to be useful for sparse matrices.

1.3 EM algorithm for matrix factorization (I. Rodin)

We want to implement expectation-maximization algorithm for finding matrix factorization $A = \Phi\Theta$, where A is matrix "term-document", Φ is matrix "word-topic" and Θ is matrix "topic-document".

In roder to find such maximization we need to solve the following optimization task:

$$\sum_{d\in D}\sum_{w\in d} a_{dw} \log\left(\sum_{t\in T} \phi_{wt}\Theta_{td}\right) \to \max_{\phi,\Theta}$$

$s.t. \phi_{wt} \geq 0; \sum_w \phi_{wt} = 1; \Theta_{td} \geq 0; \sum_t \Theta_{td} = 1$

This maximization task can be done by running EM-algorithm which is a method of simple iteration for solving the following system of equations:

$$\begin{cases} p_{tdw} = norm_{t\in T}(\phi_{wt}\Theta_{td}) \\ \phi_{wt} = norm_{w\in W}(a_{wt}), a_{wt} = \sum_{d\in D} a_{dw}p_{tdw} \\ \Theta_{td} = norm_{t\in T}(a_{td}), a_{td} = \sum_{w\in d} a_{dw}p_{tdw} \end{cases}$$

Where $norm_{i\in I}(x_i) = \frac{\max(x_i,0)}{\sum_{j\in I}\max(x_j,0)}$

This algorithm for topic modelling was implemented in [8]. We will try to write it by ourselves, add regularizations and compare algorithm results with the previous ones.

1.4 Spectral clustering (A. Akhmetgareeva)

Method is based on three basic stages:

1. Pre-processing

Firstly, construct a similarity matrix A, which represented the graph $G(V, E)$ (V - is a set of words, E is weighted with A_{ij} as defined in 1.1). In our problem each element of matrix $A_{ij} \geq 0$ equals to probability of two words i and j be in the same sentences.

2. Decomposition

Compute eigenvalues and eigenvectors of Laplacian matrix defined as

$$L^{norm} = I - D^{-1/2}AD^{-1/2},$$

where D is a diagonal matrix, $D_{ii} = \sum_j A_{ij}$ [9].

3. Partitioning

Partitioning may be done in various ways. For example, to separate into two sets (B_1, B_2) we can compute the median m of the components of the eigenvector v corresponded to second smallest eigenvalue, and placing all points whose component in v is greater than m in B_1, and the rest in B_2. The algorithm can be used for hierarchical clustering by recursively partitioning the subsets in this way. The depth of recursion is set manually (number of clusters) (we'll come up with better solution).

PageRank

Basically, PageRank problem is solve on oriented graph. We need to adapt matrix of non-oriented graph A to satisfy the problem formulation. One possible way is to set the threshold probability tp. Then build matrix W each element w_{jk} equal to 1 if the corresponding $A_{jk} \geq tp$.

Denote by p_i the importance of the i-th page. Then we define this importance as an average value of all importances of all pages that refer to the current page. It gives us a linear system

$$p_i = \sum_{j \in N(i)} \frac{p_j}{L(j)},$$

where $L(j)$ is a sum of outgoing links (elements in the j-th row of matrix W), $N(i)$ are all input neighbours. It can also be rewritten as

$$p = Gp, \quad G_{ij} = \frac{1}{L(j)}$$

or as an eigenvalue problem

$$Gp = 1p,$$

i.e. the eigenvalue 1 is already known. Note that G is left stochastic, i.e. its columns sum up to 1 [10].

2 Data

We are planning to use several sources of text data (mainly, news), however the list may decrease after we start implementing the algorithms.

2.1 LargeDatasetForKeyphrasesExtraction

This dataset was collected by [11]. It consists of pairs of files, <id>.txt with full text, <id>.key with keyphrases each on a new line

2.2 500N-KeyPhrasesCrowdAnnotated-Corpus

This dataset was collected in [12] for keywords extraction. The description follows below.

For each news article, there are 3 files: topic-id.txt - the original news article file collected from the Web were replaced by topic-id.-justTitle.txt which contains just title to avoid any potential copyright issues. We are just providing this dataset for research and educational purposes. If you believe it violates your copyright work please contact Luis to remove the document immediately.

topic-id-CrowdCountskey - the ranked list of key phrases annotated by the turkers (the number next to key phrases corresponds to the number of times out of 18-20 were selected) topic-id.key - the most and second most selected key phrases in the topic-id.CrowdCounts file.

More details about the corpus can be found in the paper (Crowdsourcing section).

2.3 FullReutersDataset

This is full news archive from Reuters collected from Jan 2007 to Aug 2016 with titles, links, and timestamps [13].

2.4 Reuters21578

Different Reuters datasets are also presented on UCI KDD, this is the dataset for text categorization from David Lewis's web-page [14].

3 Evaluation

One approach of evaluation is to simply make a "user"-based evaluation, namely when human determine the fraction of keywords correctly extracted from the text.

Chosen datasets already contains the keywords for every text, therefore we also could automatically calculate the fraction of correctly extracted keywords. Meanwhile, it is clear that we do not penalize for words which are extracted but not real keywords. What we propose, is also estimate the fraction of falsely determined keywords and use the harmonic mean of these two metrics. But of course, the evaluation metric should be strictly determined for the particular area of application.

4 Scope

Currently we are planning to work on datasets, investigate its structure and design the interfaces for our algorithms. The first prototypes will be done until the middle of December. Then we spend some time on tuning and final preparations.

Optionally, we expect that we can run the small web-application which will be able to extract the keywords using some technique among the suggested ones.

References

[1] M. Girvan and M. E. J. Newman, "Community structure in social and biological networks," *Proceedings of the National Academy of Sciences*, vol. 99, no. 12, pp. 7821–7826, 2002. DOI: 10.1073/pnas.122653799. eprint: http://www.pnas.org/content/99/12/7821.full.pdf. [Online]. Available: http://www.pnas.org/content/99/12/7821.abstract.

[2] A. Clauset, M. E. J. Newman, and C. Moore, "Finding community structure in very large networks," *Phys. Rev. E*, vol. 70, p. 066 111, 6 Dec. 2004. DOI: 10.1103/PhysRevE.70.066111. [Online]. Available: https://link.aps.org/doi/10.1103/PhysRevE.70.066111.

[3] M. E. J. Newman, "Modularity and community structure in networks," *Proceedings of the National Academy of Sciences*, vol. 103, no. 23, pp. 8577–8582, 2006. DOI: 10.1073/pnas.0601602103. eprint: http://www.pnas.org/content/103/23/8577.full.pdf. [Online]. Available: http://www.pnas.org/content/103/23/8577.abstract.

[4] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, P10008, 2008. [Online]. Available: http://stacks.iop.org/1742-5468/2008/i=10/a=P10008.

[5] D. Martin and M. Berry, "Mathematical foundations behind latent semantic analysis," pp. 35–55, Jan. 2007.

[6] D. Da Kuang, J. Choo, and H. Park, "Nonnegative matrix factorization for interactive topic modeling and document clustering," pp. 215–243, Oct. 2015.

[7] C. Boutsidis and E. Gallopoulos, "Svd based initialization: A head start for nonnegative matrix factorization," *Pattern Recogn.*, vol. 41, no. 4, pp. 1350–1362, Apr. 2008, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2007.09.010. [Online]. Available: http://dx.doi.org/10.1016/j.patcog.2007.09.010.

[8] K. Vorontsov, O. Frei, M. Apishev, P. Romov, and M. Dudarenko, "Bigartm: Open source library for regularized multimodal topic modeling of large collections," in *International Conference on Analysis of Images, Social Networks and Texts*, Springer, 2015, pp. 370–381.

[9] J. Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 888–905, Aug. 2000, ISSN: 0162-8828. DOI: 10.1109/34.868688. [Online]. Available: http://dx.doi.org/10.1109/34.868688.

[10] I. Oseledets. (). Lecture 6 eigenvalues and eigenvectors, [Online]. Available: https://github.com/oseledets/nla2017/blob/master/lectures/lecture-6.ipynb.

[11] M. Krapivin, R. Autaeu, M. Marchese, M. Krapivin, A. Autaeu, and M. Marchese, "Large dataset for keyphrases extraction," Tech. Rep., 2009.

[12] L. Marujo, A. Gershman, J. Carbonell, R. Frederking, and J. P. Neto, "Supervised topical key phrase extraction of news stories using crowdsourcing, light filtering and co-reference normalization," in *Proceedings of the LREC 2012*, 2012.

[13] P. Peremy. (). Reuters-full-dataset, [Online]. Available: https://github.com/philipperemy/Reuters-full-data-set.

[14] D. Lewis. (). Reuters21578 dataset, [Online]. Available: http://www.daviddlewis.com/resources/testcollections/reuters21578/.