PubMed Clustering - proposed solution

Dr. Jurica Ševa

January 11, 2019

1 Introduction

Here we present the proposed solution to the coding assignment part of the interview process for the position of Software Engineer - Scientific Information Retrieval at the company Invitae. The given task focuses on the use of clustering, an unsupervised machine learning (ML), where similar objects are grouped together. The notion of similarity is defined by a distance measure. To develop and evaluate the proposed solution, we are given several datasets:

1. *pmids_gold_set_labeled.txt*: PMIDs labeled with the search terms used to retrieve them. We consider this to be the ground truth.

2. *pmids_gold_set_unlabeled.txt*: ground truth PMIDS, combined and shuffled, but with labels removed.

3. *pmids_test_set_unlabeled.txt*: PMIDs with no search term labels. Note that this includes a different set of diseases from gold set and is considered the evaluation data set.

In addition, the task requires the possibility of generalization of the developed solution (i.e. applying the solution to a previously unseen set of documents and/or topics).

The proposed approach is defined through the following steps:

1. *Text pre-processing* where raw documents are converted into tokens,

2. *Feature space exploration* where the optimal feature space, defined by the properties of the ground truth, is defined,

3. *Clustering algorithms* where documents, represented in a feature space, are grouped in clusters and

4. *Evaluation* where the proposed solution is evaluated based on ground truth data set.

Each of the steps is briefly presented next.

1.1 Text pre-processing

Initially, we are given a list of PMIDs, following the format presented in pmids_test_set_unlabeled.txt. For each of the PMID values, we extract the publications' title, abstract and mesh term (unused in the solution). This is done with of *pubmed_parser*[1]. Additionally, for each PMID value, named entities (NE) available via PubTator RESTful API[2] are extracted. Prepared documents are then tokenized and stop words removed. No stemming or lemmatization is performed.

1.2 Feature space exploration

Features space exploration focused on finding the optimal way of representing documents in a high dimension space. For this purpose we use various combinations of obtained PMID descriptors: *title*, *abstract* and *NE*. We refer to this as *Input*.

In addition, three distinct *representations* are used:

- *term frequency - inverse document frequency* (tf-idf), which represent a direct mapping between input tokens and its normalized occurrence value. As such it represents a statistical measure used to evaluate how important a word is to a document in a collection or corpus. Although it is widely used, it has significant shortcomings due to its sparsity. Terms taken in consideration appeared in at least 2 documents in the corpora. Further more, l2 normalization was used to smooth obtained tf-idf values. Finally, most relevant 15% of features are kept based on the $\tilde{\chi}^2$ test.

- *Latent Semantic Analysis* (LSA), sometimes referred to as truncated SVD, or Latent Semantic Indexing (LSI), which applies singular value decomposition (SVD) to the input tf-idf matrix. As such, it enables us to address issues caused by the sparsity of tf-idf representation. The tf-idf matrix was obtained by keeping terms which appear in at least 2 documents with an additional l2 normalization applied.

- *Latent Dirichlet Allocation* (LDA) is a generative statistical framework, where each document is represented by a set of topics that are assigned to it via LDA. As such it represents a dense and probabilistic document representation, and can lead to improved performance in unsupervised ML. LDA model with 10 components yielded best performance, while an increase in the number of components leads to diminishing performance. Input values in to the LDA model were tokens counts, where each token appears in at least 2 documents in the collection.

For the tf-idf and LSA document representations, we utilize Euclidean distance. However, for LDA representations, Jensen-Shannon divergence (a symmetric variant of the Kullback-Leibler divergence) is more suitable, as we are dealing with proper probability distributions here.

To mitigate issues generated by the size of available corpora (e.g. small vocabulary leading to very sparse document representations, lack of generalization etc.), word n-grams

[1] https://github.com/titipata/pubmed_parser

[2] https://www.ncbi.nlm.nih.gov/CBBresearch/Lu/Demo/PubTator/tutorial/index.html

ranging from (1,1) to (1,4) are explored. We refer to this as *n_gram*. For each of the combinations [*text input*, *representation*] all *n_gram* values are evaluated. Finding the best document representation space, as measured by the proposed metric, is the main focus of this work.

1.3 Clustering algorithms: choice and evaluation

As the goal of the proposed approach is to generate a solution that will generalize well, it is imperative that the number of clusters is not a parameter of the used clustering algorithm. Therefore, HDBSCAN[3] has been selected as the clustering algorithm. Euclidean distance is used for the tf-idf and LSA representation while Jensen-Shannon is used when working with LDA representation of documents. HDBSCAN is a density based clustering algorithm which does not require pre-defining the number of target clusters. As such it works well with the approach presented here. Additionally, it uses the notion of density, defined with a parameter ϵ, and as such captures clusters which are arbitrarily shaped. Underlying it uses a hierarchical clustering approach, which enables efficient clustering. As a comparison, the same evaluation was performed with the K-means[4] clustering algorithm, with k set to the number of unique topics in ground truth (6). Although this presents a limitation on the proposed approach, it was included for completeness and comparability. We expect K-means to outperform HDBSCAN. No hyper-parameter optimization was performed on either of clustering approaches. To measure the performance of the proposed approach, and subsequently, choose the best model, three measures are used: a) *Cumulative*, which maximizes the sum of distances between individual clusters and minimizes the maximum intra-cluster distance of individual cluster points to cluster centroid, b) *Homogeneity score* (HS) and c) *Adjusted Mutual Information* (AMI).

The performance of individual feature space is first evaluated using the Cumulative measure. This ensures that the best features space is the one where the cluster has points closest to the centroid while centroids are furthest apart. Cluster centroids are calculated as the mean value of all documents assigned to individual clusters. The results point to several conclusions. First, one could wrongly conclude that the use of title or (title, NE) as input elements to describe individual document as it produces the biggest Cumulative score in all three tested feature representations. This performance can easily be attributed to the size of the ground truth corpora (only 103 documents!). When the number of documents increases, the performance of this approach would most likely decrease (considerably). This does, however, point to the fact that the title is a discriminative input element, and should be included in the document descriptors. For the other three input combinations, (title, abstract, NE) shows best performance on two out of three representation schemes. Same as title, NE is also a strong discriminative input. As they are already present as tokens in the abstract this represents a redundant input, but one which can further strengthen the generated clusters. It is recommended that abstract, the most diverse input, is part of the document description as it enables a broader generalization than with (title, NE) input combination. An additional, currently not implemented, option would include the species and

[3]https://hdbscan.readthedocs.io/en/latest/index.html

[4]https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

			K-Means		HDBSCAN		
Representation	Input	n_gram	HS	AMI	HS	AMI	Cumulative
LDA	abstract	1_4	0.305	0.227	0.074	0.044	6.465
	title	1_4	0.784	0.728	0.763	0.719	14.723
	title_NE	*1_4*	*0.746*	*0.713*	*0.590*	*0.561*	*18.407*
	title_abstract	1_4	0.303	0.228	0.097	0.049	6.230
	title_abstract_NE	1_3	0.339	0.271	0.131	0.069	5.754
LSA	abstract	1_4	0.831	0.789	-0.000	-0.000	-10.207
	title	1_1	0.921	0.897	0.458	0.419	-0.813
	title_NE	*1_1*	*0.977*	*0.975*	*0.967*	*0.950*	*1.580*
	title_abstract	1_4	0.894	0.824	0.940	0.851	-9.055
	title_abstract_NE	1_3	0.941	0.872	1.000	0.980	-7.038
tf - idf	abstract	1_4	0.953	0.949	-0.000	-0.000	-1.410
	title	*1_1*	*0.909*	*0.880*	*0.867*	*0.636*	*9.569*
	title_NE	1_2	1.000	1.000	0.883	0.816	9.139
	title_abstract	1_4	0.941	0.936	0.492	0.475	-0.276
	title_abstract_NE	1_3	0.899	0.863	0.580	0.558	1.326

Table 1: Best performing feature space for distinct representation schemes and input elements, evaluated based on the Cumulative measure. LSA representation fails to find clusters of documents as present in ground truth. LDA finds most coherent clusters (as present in ground truth). *Italic* rows represent best performing [Representation, Input, n_gram] combination. In all cases most discriminative features are members of best performing combination.

the normalized IDs combined with respective entity types found in abstracts. This information is easily found in PubTator pre-annotated PubMed corpus[5]. One can also see that LSA and tf-idf representation schemes clearly outperform LDA. Again, this should not be taken as a certainty due to the probabilistic nature of LDA and the size of the ground truth corpora. With the increase of corpora size, LDA would most likely outperform. As far as the n-grams tested, almost all values in Table 1 are associated with four-grams and to a lesser extent uni- and tri-grams. Bi-grams are hardly represented. Generating additional features with respect to various n-gram combinations proves to be the better option.

As the *Cumulative* score greatly varies between the three developed representation schemes, a second optimization approach, based on HS and AMI, is evaluated. Evaluation results, optimized against the $max \sum(AMI_i, HS_i)$ measure are presented in Table 2. Not surprisingly, K-means outperforms HDBSCAN on (almoast) all combinations of (Representation, Input, n_gram). HDBSCAN comes closest, and outperforms K-means with the LSA representation, with title as input, the smallest feature space. As the size of the feature space grows, the performance of K-means does not (largely) vary. tf-idf and LSA representations, with (title, abstract) and (title, abstract, NE) input show the best performance in both clustering approaches. LDA underperforms once again; this is again contributed to the probabilistic nature of the approach and the size of the given corpora. Although results presented in Table 2 give confidence in the quality of the proposed solution, it is possible that it will not

[5]ftp://ftp.ncbi.nlm.nih.gov/pub/lu/PubTator/bioconcepts2pubtator.gz

perform adequately well on unseen documents and topics.

		K-Means			HDBSCAN		
Representation	Input	n_gram	HS	AMI	n_gram	HS	AMI
LDA	abstract	1,4	0.305	0.227	1,1	0.155	0.060
	title	1,4	0.784	0.728	1,4	*0.763*	*0.719*
	title,NE	1,1	*0.812*	*0.768*	1,1	0.647	0.605
	title,abstract	1,4	0.303	0.228	1,1	0.106	0.059
	title,abstract,NE	1,4	0.348	0.281	1,3	0.131	0.069
LSA	abstract	1,2	0.851	0.814	1,1	-0.000	-0.000
	title	1,1	0.921	0.897	1,2	0.611	0.583
	title,NE	*1,2*	*1.000*	*1.000*	*1,2*	*1.000*	*1.000*
	title,abstract	1,2	0.897	0.824	1,3	0.940	0.851
	title,abstract,NE	1,1	0.931	0.920	**1,2**	**1.000**	**0.980**
tf-idf	abstract	1,4	0.953	0.949	1,1	-0.000	-0.000
	title	1,1	0.909	0.880	*1,2*	*0.904*	*0.729*
	title,NE	*1,1*	1.000	1.000	1,1	0.941	0.877
	title,abstract	1,2	0.951	0.947	1,3	0.492	0.475
	title,abstract_NE	**1_1**	**1.000**	**1.000**	1_3	0.580	0.558

Table 2: Evaluation of clustering results with optimized feature space and different evaluation. Best performing model with (title, abstract, NE) input is used in production. *Italic* rows represent representation-based best performing models. **Bold** rows represent models used for inference.

2 Discussion: proposed solution and design choices

This report presents the proposed solution for the problem of document clustering. The task was constrained by the requirement of obtaining clusters most similar to the ground truth labels. With that in mind, the problem description has omitted information describing, among other, the full document set assigned to a topic, ranking of given documents as well as retrieval approach used in obtaining the documents. Therefore, although there is a notion of ground truth, we can not be confident in its completeness. Further more, we are given only 103 documents, unevenly distributed over 6 distinct topics.

In the proposed approach, we focused on creating an optimal future space, consisting of a triple [*Representation, Input, n_gram*]. For this purpose, two distinct optimization approaches have been presented and evaluated. First, *Cumulative*, focuses on optimizing the future space by increasing the distances between centroids while minimizing the distances found between points and centroid of individual clusters. The second approach, $max \sum_{i=0}^{n}(AMI_i, HS_i)$, relies on the other hand heavily on the given notion of ground truth. As such, it encompasses bias towards the given ground truth and might poorly generalize on unseen corpora/topics. Nevertheless, best performing models as defined by this approach have been used for inference. Finally, there are several possible extensions to this work.

First, one could use a Random Forrest supervised classification model to learn discriminative features. As we focus on (fully) unsupervised approach to obtain most discriminative features, and due to time constraints, we discarded this option for this work. Additionally, (variational) AutoEncoder could be used to learn a low level latent document representation, which is then used as input to the clustering algorithm of choice. Although this would fit our focus on fully unsupervised feature engineering, this too was discarded due to time constraints.

The developed pipeline is an object-oriented pipeline, mostly built on top of existing libraries (NLTK, scikit-learn and HDBSCAN among others). The main modules are:

- **TokenizePreprocessor**, which tokenizes documents into a set of tokens. It has a built in reg-exp matching for token normalization. This feature, however, was not used as it would negatively impact the size of vocabulary which would, as a consequence, have a negative impact on the sparseness of the representations (i.e. it would increase it).

- **preprocess_text**, where given PMID values are extended with available information from PubMed as well as PubTator service. In addition, it prepares and serializes extended corpora for faster future operation, removes stop words and transforms raw text in one of the proposed representations. It uses TokenizePreprocessor to perform tokenization.

- **Exploration**, which performs feature selection, hyper-parameter optimization (where applicable), model evaluation and serialization. Additionally, it is used for inferring clusters on new corpora using serialized models.

- **main** which presents the command line interface. No GUI interface has been developed due to the complexity of the problem and time constraints given.

The implementation, as it is, can be used with any number of documents/topics. It supports multiprocessing (where applicable). Due to given time constraints and the size of the implementation, computational complexity is currently not given.