Document retrieval and question answering in medical documents.
A large-scale corpus challenge.

Eric Curea
Research Institute for Artificial Intelligence
"MIHAI DRAGANESCU",
Romanian Academy
eric@racai.ro

Abstract

Whenever employed on large datasets, information retrieval works by isolating a subset of documents from the larger dataset and then proceeding with low-level processing of the text. This is usually carried out by means of adding index-terms to each document in the collection. In this paper we deal with automatic document classification and index-term detection applied on large-scale medical corpora. In our methodology we employ a linear classifier and we test our results on the BioASQ training corpora, which is a collection of 12 million MeSH-indexed medical abstracts. We cover both term-indexing, result retrieval and result ranking based on distributed word representations.

1 Introduction

Automatic key-wording is the process of enriching text documents with pre-specified classes (topics or themes). The primary motivation is that in information retrieval one can easily use these keywords for automatically filtering and obtaining a subset of documents form a large-scale corpus, documents that share common traits linked to their domain, topic, title, publication source, authors, etc. As such, automatic key-wording and document indexing (based on these keywords) helps people to find information in huge resources.

Currently, most of the on-line information is available in the form of unstructured documents and this is unlikely to change in the foreseeable future. Though, several initiatives to force users into manually labeling their on-line publications using specialized markup have been proposed (one good example is Google Markup Language[1]), scientific publications are unlikely to be subject to such annotations, mainly because they employ printable formats such as Postscript and PDF (which, in fortunate situations, can be converted into plain text).

Thus, NLP task such as unsupervised document clustering represents a key-task in information retrieval. Due to the increased availability of documents in digital form and the ensuing need to access them in flexible ways, content-based document management tasks (collectively known as information retrieval IR) have gained a prominent status in the research community in the past decade. The task of Document classification or document categorization, the activity of labeling natural language texts with thematic categories from a predefined set, is very important and still evolving thanks to increased applicative interest and to the availability of more powerful hardware.

To accomplish the task of document classification, an increasing number of computational and statistical approaches have been developed over the years, to mention a few: Suport Vector Machines (SVMs) (Manevitz and Yousef, 2001; Joachims, 1998), maximum entropy (Ratnaparkhi, 1998; El-Halees, 2015), word-distributional clustering (Baker and McCallum, 1998), weighted K-Nearest-Neighbor classification (Han et al., 2001; Larsen and Aone, 1999), linear classifiers (Lewis et al., 1996), Naive Bayes methods (McCallum and Nigam, 1998), artificial neural networks (Zhang and Zhou, 2006; Collobert and Weston, 2008; Lai et al., 2015), decision trees (Lewis and Ringuette, 1994).

Our work is focused on automatic labeling of medical text, using Medical Subject Headings (MeSH)[2] terms (Rogers, 1963) (see section 3)

[1] https://developers.google.com/search/docs/guides/intro-structured-data - accessed 2017-05-18

[2] https://www.ncbi.nlm.nih.gov/pmc/articles/PMC35238/

Proceedings of the Biomedical NLP Workshop associated with RANLP 2017, pages 1–7,
Varna, Bulgaria, 8 September 2017.
https://doi.org/10.26615/978-954-452-044-1_001

and information retrieval for question answering based on the analysis of article abstracts (see section 4). The training, evaluation and test datasets used in the validation of our procedure are part of the BioASQ[3] (Tsatsaronis et al., 2012) evaluation campaign.

2 Corpus description

The train set corpus contained articles from the free on-line repository PubMed [4]

The training data is composed of a very large number of documents collected from PubMed, which have been semi-automatically annotated with MeSH terms, with the help of human curators. Aside from the MeSH terms, each entry in the dataset contained important meta-data such as: the title of the paper, the journal where the paper was published, publishing year and the paper's abstract.

The training set is JSON-encoded and contains the following fields for each article:

1. pmid : An unique identifier assigned to each paper - used for internal evaluation purposes;
2. title : The original title of the article
3. abstractText : the abstract of the article,
4. year : the year the article was published,
5. journal : the journal the article was published, and
6. meshMajor : a list with the major MeSH headings of the article.

For clarity, we also provide an excerpt from the training data, presenting the structure af each article collected in the large-scale corpus:

```
{"articles": [{"journal":"journal..","
    abstractText":"text..", "meshMajor
    ":["mesh1",...,"meshN"], "pmid":"
    PMID", "title":"title..", "year":"
    YYYY"},..., {..}]}
```

To offer a better view over the training data, we must specify that the total number of articles is 12,834,585, published in over 9,000 journals, with an average of 1,421.64 articles in each journal, published from 1946 to 2016 with most articles (over 600,000) selected from 2014, a distribution

Table 1: Label distribution over training data

labeled documents	distribution	percentage
>1,000,000	10	0.04%
500,000 1,000,000	7	0.03%
100,000 500,000	137	0.49%
50,000 100,000	223	0.80%
10,000 50,000	2,240	8.07%
1,000 10,000	10,053	36.20%
<1,000	15,103	54.38%
total	27,773.00	

of 12.66 average MeSHes per article, going from the MeSH "humans" with an occurrence of over 8 millions to MeSHes like "tropaeolaceae" that only occur once, yielding a MeSH coverage of 27,773 MeSHes composed either from a single word of a construct like "magnetic_resonance_imaging". All this in a total of 20.5GB (plain/text) and 6.29GB (compressed data). Table 2 provides generic information regarding frequent versus uncommon MeSHes, while table 1 captures the "spread" of the MeSHes throughout the training data.

As can easily be seen from table 1, 10 of the frequent MeSHes like "humans", "male", "female" or "animals", are used to label more the 1M documents, only 7 fall within the 500K-1M range and 360 between 50K and 500K (we further refer to them as category A). On the opposite side, 2K MeSHes are found in 10K-50K documents, 10K MeSHes in 1K-10K documents and more than 15K MeSHes have an occurrence of less than 1K (category B). The high occurring MeSHes (category A) represent less than 2% of the total number of labels, which indicates that in most cases any ML system will most likely not be able to model the rest of 98% of the labels based on this corpus. To clarify our previous statement, it is expected that most classifiers will have a small recall for 98% of the labels, mainly because the objective of minimizing the "overall" accuracy is easily achieved by preferring not to emit any label rather than incorrectly classifying documents with bad labels and only for less than 2% of the total number of labels the systems will have a chance of a high recall.

3 Automatic MeSH labeling

Currently, there are 28,489 descriptors in MeSH 2017 that were used in the creation of the training data. However, due to the unbalanced occurrence

- accessed 2017-05-18

[3]http://bioasq.org - last accessed 2017-05-09

[4]https://www.ncbi.nlm.nih.gov/pubmed/ - accessed 2017-04-29

Table 2: MeSH distribution

ID	MESH	count	ID	MESH	count
1	humans	8,103,280	27	kinetics	366,997
2	male	5,351,269	28	cell_line	331,436
3	female	5,169,536	29	surveys_and_questionnaires	316,552
4	animals	3,932,184	30	rna+messenger	314,638
5	adult	3,119,705	31	dose-response_relationship+drug	313,386
6	middle_aged	2,782,688	32	reproducibility_of_results	285,023
7	aged	1,936,405	33	infant+newborn	283,249
8	adolescent	1,219,944	34	mutation	278,419
9	rats	1,116,126	35	united_states	272,593
10	mice	1,045,215	36	brain	269,598
11	child	826,020	37	rats+sprague-dawley	265,472
12	time_factors	793,584	38	sensitivity_and_specificity	264,091
13	aged+80_and_over	636,261	39	prognosis	259,335
14	molecular_sequence_data	590,276	40	in_vitro_techniques	258,033
15	treatment_outcome	571,489	41	age_factors	254,441
16	retrospective_studies	547,781	42	liver	248,866
17	child+preschool	510,539		.....	.....
18	young_adult	494,101		ephemerovirus	5
19	risk_factors	450,495		.....	.....
20	follow-up_studies	447,572		zigadenus	4
21	cells+cultured	428,059		.....	.....
22	amino_acid_sequence	395,146		cytophagaceae_infections	3
23	prospective_studies	394,813		.....	.....
24	pregnancy	392,281		duboisia	2
25	infant	387,000		.....	.....
26	base_sequence	385,031		childhood-onset_fluency_disorder	1

of terms combined with the large scale of the corpus, we ran our experiments on a smaller sub-set of MeSH terms, composed of only 154 most frequent items.

In the classification process we took into account as much information as we can and have access to, about each document in the large-scale corpus. The title of a document usually holds key information about the content of the document. The journal in which it was published is likely to carry weight in the label assigning process as only specific types of documents can be published in certain types of journals. The year in which the document was published will tell the system if the information retrieved from the document has a chance of not being up to date or it might be completely outdated and superseded by more recent research, in which case the system should at least try to see if newer publications might hold better results or more important supplementary information. The abstract text is the place where the system can spend most processing time and apply as many tests, approximations and refinements, because this is the place where most articles condense the biggest amount of relevant information about the content of the document. Of course finding possible relevant information in the abstract text is only part of the equation. The more important part is determining relevant relations between different relevant lexical tokens, the location of the information segments, distance between the different relevant lexical tokens inside the abstract, number of occurrences, similarity to the information determined in the question (W2V, cosine similarity(Steinbach et al., 2000)).

All the input features were treated in a bag-of-words manner, from which we removed any feature (word) with an occurrence rate lower than 100. This threshold of 100 was selected after testing different limits that yielded either too few features left to test with or too low occurrence rate for the feature to be relevant. Initially, our training data contained 7,466,119 unique features and the pruning process reduced this number to

only 123,255. For the classification task we employed an ensemble of linear classifiers. Each possible output MeSH was associated with a classifier, which was trained in a 1-vs-all style to predict if the system should or should not assign that label, based on the input features. The output of the linear model ranged from -1 (do not assign a label) to 1 (assign a label) and was computed using Equation 1, with w computed using the delta-rule (Equation 2):

$$y = \sum_{1}^{n} w_n \cdot x_n \tag{1}$$

$$\Delta w_k = \alpha \cdot (t - y) \cdot x_k \tag{2}$$

where
y is the output of the classifier
t is the desired output of the classifier (-1 or 1)
x_i is the ith input feature
w_i is the weight of the i-th input feature
α is the learning-rate (set to 10^{-3})

When we trained our ensemble of classifiers we divided our training data into 9/10 for training and 1/10 for development, while trying to preserve as best as possible the initial distribution for each of the labels in both sets. Training was done iteratively (compute new value for w using the training set and measure accuracy on the development set) and the stopping condition was not to have any improvements on the development set for more than 20 iterations. At the end of the training process we kept the w that achieved the highest accuracy on the development set.

Table 3: Labeling results

System	MiP	MiR	Acc.
Sequencer	0.0920	0.0964	0.0494
Default MTI	0.6148	0.6286	0.4594
Our System	**0.7681**	**0.1472**	**0.1381**
DeepMeSH4	0.6671	0.6289	0.4839
MZ1	0.6495	0.3985	0.3299
DeepMeSH3	0.6898	0.6170	0.4877
DeepMeSH2	0.6895	0.6432	0.5059
DeepMeSH1	0.7025	0.6282	0.5025
DeepMeSH5	0.7198	0.6122	0.5024

Table 3 shows the accuracy (Acc), Micro Precision (MiP) and Micro Recall (MiR) of our system, measured on one of the datasets. It also offers a comparative view between our methodology and the other systems present in the competition. We must mention that the overall performance figures are measured using all the available MeSHes, not the pruned subset.

4 Result ranking

For this we take each lexical component of the key set of data extracted from the corpus and we try to find if the classified documents from the corpus approximate to possible synonyms of lexical component. For each lexical component of the key set of data extracted from the question, we calculated a list of lexical elements that can be considered similar in meaning using "cosine similarity" computed over distributed word representations (Mikolov et al., 2013). The vectors (100-dimensional) were computed using the word2vec[5] tool on a specific subset of Wikipedia combined with additional raw text resources provided as part of the BioASQ challenge. In order to compile the subset from Wikipedia we followed a simple bootstrapping procedure:

1. We downloaded the latest Wikipedia XML Dump at that date from the official web-site, on which we run a version of WikipediaExtractor[6], that was modified to preserve categories;

2. We seeded a list of categories, using the first level of categories on the Wikipedia site for the "Biomedical" main category;

3. We iterated 3 times through the entire corpus and we consolidated our category list, by adding categories that were associated with our initial category list, each time updating our seeded list;

4. We kept all documents that had at least one category from our final category list.

Given a "question" our IR process is: (a) we extract a list of keywords from the query, by removing function words from using a predefined dictionary; (b) we use the keywords to retrieve the top 1M documents from the initial corpus; (c) we re-rank our results and obtain a list with the top-10 most relevant documents. Document ranking

[5]https://github.com/dav/word2vec - accessed 2017-04-05
[6]https://github.com/bwbaugh/wikipedia-extractor - accessed 2017-01-28

Table 4: Test-set results

System Name	Mean precision	Recall	F-measure	Map	GMAP
Top 100 Baseline	0.2460	0.2845	0.1333	0.1606	0.0028
Top 50 Baseline	0.2470	0.2591	0.1920	0.1503	0.0024
fdu_5b	0.1865	0.2228	0.1791	0.1300	0.0084
Our System	**0.4000**	0.2222	**0.2857**	0.1238	**0.1238**
MCTeamMM	0.2266	0.1481	0.1249	0.0892	0.0005
MCTeamMM10	0.0326	0.1481	0.0436	0.0892	0.0005
Wishart-S1	0.0465	0.0484	0.0350	0.0237	0.0001

Figure 1: Distribution of publications each year

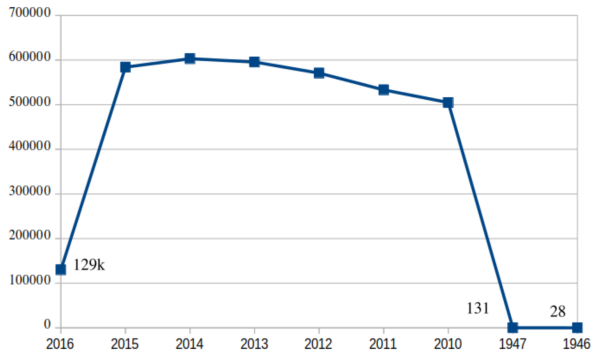
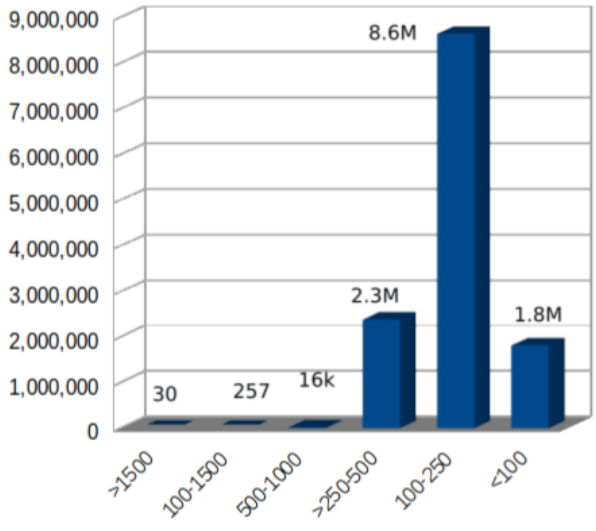


Figure 2: Distribution of words in articles



is performed using Equation 3, which is designed to take into account keyword synonymic coverage, but currently ignores synonymic frequencies in the text (in our empirical experiments we found that introducing this factor decrease the overall precision of the system - in our opinion, mainly because word-embeddings are prone to capturing contextual similarities, rather than actual synonymic behavior).

$$S_d = \frac{1}{k} \cdot \sum_{i=1}^{k} \max_{j=1}^{m}(cos(t_i, d_j)) \qquad (3)$$

where
S_d - is the relevance of document d
k - is the number of keywords in the query
m - is the number of words in the document
t_i - is the word embedding for term i in the query
d_j - is the word embedding for term j in the document

Table 4 shows the precision, recall and F-score of our system, measured on one of the datasets. It also offers a comparative view between our methodology and the other systems present in the competition. We must mention that the overall performance figures are measured using all the available MeSHes, not the pruned subset.

5 Snippets

Usually not all the text in the retrieved abstract is part of a good answer to a given question. So finding the most relevant, shortest part of the abstract was nest step.

To approximate the shortest span of text in each abstract of the documents, that represents the best response to the question, we selected a list of all the lexical tokens in the abstract text that correspond or might have generated the relevant label. At first glance, the snippet would be starting from the beginning of the first sentence that contains a

token from the list and finishing at the end of the last sentence that contains a token from the list.

Of course this list has a high probability of having duplicates. These duplicates have no value for detecting the shortest relevant text. So we calculate from the current abstract, the shortest span of text that still contains all of the lexical tokens but we ignore any duplicates in the list.

To help explain the previous statement we will use the following example:

```
"document": .... [token_1]....[token_1
    ]...[token_2].....[token_1]....[
    token_3].....[token_4].....[token_5
    ]....[token_1] ...
```

It can easily be seen in the example that the first iteration of "token_1" holds no value for the purpose of finding the shortest relevant span of text an neither does the second iteration even though it is position in closer proximity to another token from the list. The list is not in any way ordered so the placement of the second token: "token_2" in front of the first token "token_1" is irrelevant. The existence of a different token in front of the current token: "token_2" before "token_1" only means that this iteration of "token_1" is a viable candidate for the shortest relevant span of text. Finally the final iteration of "token_1" has no other tokens placed after it so we considered this iteration to hold less value for a snippet. No other token had a duplicate in this example so in this case the shortest most relevant span of text was:

```
"snippet": [token_2].....[token_1]....[
    token_3].....[token_4].....[token_5]
```

It is worth noting that there were of course cases when the system would present the snippet as being the same as the entirety of the abstract text.

6 Conclusions and future work

In this article we presented a "biomedical" oriented system that automatically assigns MeSH labels to documents in a large-scale corpus. Our apprach is based on a linear classifier, trained in a 1-vs-all style for each possible MESH.

The system then retrieves answers from said corpus for questions relevant to the medical field. Each question yields a number of "n" best ranked documents that relate to the question. We achieve this by first selecting the relevant lexical tokens from the questions. Then we use Word2Vec for 100 length vectors in order to calculate the cosine similarity to approximate "x" closest lexical concepts for each of the tokens from the question.

Our system also provides a corresponding list of "n" snippets from the best ranked documents, the shortest span of text which contain the information from the abstract most relevant for the current question. This is done by discarding any sentence from the abstract text that does not contain any token from a determined list or only contains low relevance duplicates of tokens from said list.

Currently we do not deal with determining and extracting lexical dependencies between words and we only focus on relevant-document retrieval. However, our future development plans include extending our system to be able to answer yes/no, factoid and item-list questions. Additionally we plan to include multilingual data from various sources and investigate cross-lingual techniques for document retrieval and machine translation for delivering the cross-lingual results in the user's native language.

References

L Douglas Baker and Andrew Kachites McCallum. 1998. Distributional clustering of words for text classification. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 96–103. ACM.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167. ACM.

Alaa M El-Halees. 2015. Arabic text classification using maximum entropy. *IUG Journal of Natural Studies*, 15(1).

Eui-Hong Sam Han, George Karypis, and Vipin Kumar. 2001. Text categorization using weight adjusted k-nearest neighbor classification. In *Pacific-asia conference on knowledge discovery and data mining*, pages 53–65. Springer.

Thorsten Joachims. 1998. Text categorization with support vector machines: Learning with many relevant features. *Machine learning: ECML-98*, pages 137–142.

Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Recurrent convolutional neural networks for text classification. In *AAAI*, volume 333, pages 2267–2273.

Bjornar Larsen and Chinatsu Aone. 1999. Fast and effective text mining using linear-time document clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–22. ACM.

David D Lewis and Marc Ringuette. 1994. A comparison of two learning algorithms for text categorization. In *Third annual symposium on document analysis and information retrieval*, volume 33, pages 81–93.

David D Lewis, Robert E Schapire, James P Callan, and Ron Papka. 1996. Training algorithms for linear text classifiers. In *Proceedings of the 19th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 298–306. ACM.

Larry M Manevitz and Malik Yousef. 2001. One-class svms for document classification. *Journal of Machine Learning Research*, 2(Dec):139–154.

A McCallum and K Nigam. 1998. A comparison of event models for naive bayes text classification; 1998. *Disponıvel em:¡ citeseer. nj. nec. com/mccallum98comparison. html*.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Adwait Ratnaparkhi. 1998. *Maximum entropy models for natural language ambiguity resolution*. Ph.D. thesis, University of Pennsylvania.

FB Rogers. 1963. Medical subject headings. *Bulletin of the Medical Library Association*, 51:114–116.

Michael Steinbach, George Karypis, Vipin Kumar, et al. 2000. A comparison of document clustering techniques. In *KDD workshop on text mining*, volume 400, pages 525–526. Boston.

George Tsatsaronis, Michael Schroeder, Georgios Paliouras, Yannis Almirantis, Ion Androutsopoulos, Eric Gaussier, Patrick Gallinari, Thierry Artieres, Michael R Alvers, Matthias Zschunke, et al. 2012. Bioasq: A challenge on large-scale biomedical semantic indexing and question answering. In *AAAI fall symposium: Information retrieval and knowledge discovery in biomedical text*.

Min-Ling Zhang and Zhi-Hua Zhou. 2006. Multilabel neural networks with applications to functional genomics and text categorization. *IEEE transactions on Knowledge and Data Engineering*, 18(10):1338–1351.