An Approach to the Problem of Annotation of Research Publications

Ekaterina Chernyak
National Research University – Higher School of Economics
Moscow, Russia
echernyak@hse.ru

ABSTRACT

An approach to multiple labelling research papers is explored. We develop techniques for annotating/labelling research papers in informatics and computer sciences with key phrases taken from the ACM Computing Classification System. The techniques utilise a phrase-to-text relevance measure so that only those phrases that are most relevant go to the annotation. Three phrase-to-text relevance measures are experimentally compared in this setting. The measures are: (a) cosine relevance score between conventional vector space representations of the texts coded with tf-idf weighting; (b) popular characteristic of probability of term generation BM25; and (c) an in-house characteristic of conditional probability of symbols averaged over matching fragments in suffix trees representing texts and phrases, CPAMF. In an experiment conducted over a set of texts published in journals of the ACM and manually annotated by their authors, CPAMF outperforms both the cosine measure and BM25 by a wide margin.

Categories and Subject Descriptors: I.2.7 Natural Language Processing: Text analysis

Keywords: phrase-to-text relevance; annotated suffix tree; text labelling

1. INTRODUCTION

Automating semantic text analysis is currently one of the most important and demanding issues in informatics and computer science. Many researches currently are engaged in text classification or categorisation problems. Generically, the problem can be stated as follows. Given a collection of text documents and a set of labels represented by a text phrase each, annotate a document with relevant labels. This problem underlies projects in information retrieval [?], text cataloging [?, ?], text annotation [?], etc. Both machine learning approaches, self-learning and supervised learning, have been followed in the literature. Initially, an assumption was that a text document should be annotated by a single category, like in a classifying task. Currently, a more relaxed assumption of using several labels (multi-label classification) is getting prevailed. In our case, the issue is in annotation of research papers, so that a taxonomy of the corresponding domain should be utilized. Specifically, papers in Computer Science can be categorized according to Computing Classification System developed by the international Association for Computing Machinery. CCS is a hierarchical taxonomy in which every topic comprises some finer topics and itself is part of more general topic. For example, according to ACM CCS, "Data mining" is part of "Information systems applications" and it is divided in "Clustering", "Collaborative filtering", "Association rules" and etc. Authors of [?] review and empirically compare supervised methods for multi-label classification, and paper [?] applies some of these methods to ACM CCS.

Self-learning labelling methods so far have received little attention, perhaps because both proper methods and test collections are lacking. In general self-learning methods of text analysis lead to rather poor results in comparison to supervised methods (see, for example [?, ?]). This work presents a study on usage string-to-text relevance measures for self-learning multi-label annotation of research papers. String-to-text relevance measures are widely used in various text analysis tasks [?, ?, ?, ?]. They involve only sequences of characters and their frequencies and disregard syntax, grammar, and semantic features. This makes this approach language independent, thus much universal. On the other hand, loosing natural language features makes it difficult or impossible to exploit important semantic relations, such as synonymy, and of course the sentence structure.

The goal of this paper is to experimentally compare string-to-text relevance measures as tools for self-learning multi-label classification of research papers. We take most popular measures from the vector space [?] and probabilistic models [?]. As to the suffix tree approach [?, ?] we extend it with a relevance measure of our own [?]. Our measure, CPAMF, has a clear operational meaning as a characteristic of conditional probability of symbols averaged over matching fragments of strings and texts .

A very first issue one faces when conducting experiments on text annotation is in finding a reliable test bed. To this end, we use a collection of abstracts of papers published in journals of ACM and manually annotated by the authors of papers with key phrases taken from ACM CCS 2012 taxonomy. We apply several most popular text preprocessing techniques to this collection. To measure the quality of obtained annotations we use two popular measures and a measure of our own.

WSDM'15, February 2–6, 2015, Shanghai, China.

http://dx.doi.org/10.1145/2684822.2697032.

The paper is organised as follows. Section 2 defines relevance measures to be used in the experiments. Section 3 lists the text preprocessing techniques under consideration. Section 4 presents the experimental setup. Section 5 describes three measures to evaluate how good are found annotations. Section 6 presents results of the experimental comparison of text preprocessing techniques and relevance measures. Section 7 gives a conclusion. The work was partly supported by the group project “Methods for text analysis and visualization” funded by the NRU HSE Academic Fund in 2013-2014 and supervised by prof. Boris Mirkin.

2. RELEVANCE MEASURES

Consider three models of text representation: vector space model (VSM), probabilistic model (PM) and annotated suffix trees (AST) and corresponding phrase-to-text relevance measures. The relevance measures are used to annotate research papers. Each paper is represented by its abstract and a set of taxonomy topics (index terms), manually assigned by the authors of the paper. The better the computational annotations match those manual, the better the labelling algorithm.

2.1 Vector space model

According to [?], a text document is represented as a set of words (or any other document constituents). Each word corresponds to a dimension in the vector space. A vector component is usually the tf-idf weight, which stands for the frequency of the word in the document divided by the logarithm of the relative number of documents containing the word [?]. The phrase-to-text relevance is defined in the following way: represent both the phrase and the text as vectors in the same vector space. Then the relevance value is computed as the similarity between the vectors.

Let us consider the abstract of a research paper and the ACM CCS taxonomy topic as two sets of terms, where the term is either a word as it occurs in the text or a preprocessed word. The tf-idf weight of the term is $w_{ia} = tf \times idf = tf_{ia} * \log \frac{|A|}{n(t_i)+1}$, where tf_{ia} is the frequency of the term t_i in the abstract a, $n(t_i)$ is the number of abstracts, containing the term t_i, $|A|$ is the total number of abstracts.

We construct a corresponding vector in the space of terms for each abstract. The number of dimensions is N, the components are tf-idf weights. In the same way we construct vectors of tf-idf weights for taxonomy topics. We estimate the taxonomy topic-to-abstract relevance as the cosine between corresponding vectors. Let w_{ia}, w_{iq} be the tf-idf weights of the t_i for the abstract $a \in A$ and taxonomy topic q. The cosine relevance measure is as follows:

$$relevance(topic, abstract) = \cos \vec{q}\vec{a} = \frac{\vec{a} \times \vec{q}}{||\vec{a}|| \times ||\vec{q}||} = \\ = \frac{\sum_{i=1}^{N} w_{ia} \times w_{iq}}{\sqrt{\sum_{i=1}^{N} w_{ia}}\sqrt{\sum_{i=1}^{N} w_{iq}}}$$

2.2 Probabilistic model

The probabilistic model is built under a theoretical assumption that any text under consideration is a mixture of two Poisson distributions. One distribution is responsible for ordinary words, another for so called elite words. The document is in a sense about the concept expressed with elite words [?]. A phrase-to-text relevance measure in this model estimates the probability that the words that occur in the phrase are elite. Let us consider both the abstract and a taxonomy topic as two sets of terms. The BM25 measure is defined as follows:

$$relevance(topic, abstract) = \sum_{i=1}^{N} IDF(t_i) \times \\ \times \frac{k_1 + 1)tf_{ia}}{tf_{ia} + k_1(1 - b + b\frac{|A|}{avgdl})}$$

where $avgdl$ is the average number of words in an abstract, b, k_1 are constant numbers, taken to be equal to 1.5 and 0.75, respectively, according to [?]. The IDF function is used as a normalizing factor:

$$IDF(t_i) = \log \frac{|A| - n(t_i) + 0.5}{n(t_i) + 0.5},$$

where $|A|$ is the total number of annotations and $n(t_i)$ is the number of abstracts containing the term t_i. The IDF function stands for the inverse frequency: the more abstracts contain this term, the less important it is.

2.3 Annotated suffix tree

According to the annotated suffix tree model [?, ?], a text document is not a set of words or terms, but a set of the so-called fragments, the sequences of characters arranged in the same order as they occur in the text. Each fragment is characterized by a float number. The greater the number is, the more important the fragment is for the text. An annotated suffix tree (see Fig. 1) is a data structure used for computing and storing all fragments of the text and their frequencies. It is a rooted tree in which:

- Every node corresponds to one character
- Every node is labeled by the frequency of the text fragment encoded by the path from the root to the node.

To build an AST, we split the text in strings of three words, and apply them consecutively to warrant that the resulting AST has a modest size. Moreover, it is good for matching with taxonomy topics because they are phrases of similar length. Our algorithm for constructing an AST [?] is a light modification of the well-known algorithms for constructing suffix trees [?], [?]. First we build an AST for every abstract. Next we match the taxonomy topics to the AST to estimate the relevance. This is done in several steps:

1. Every taxonomy topic is split in suffixes;
2. Every suffix is matched to the AST. A match is a path from the root of the AST, that coincides with the beginning of the current suffix of the whole suffix. To estimate the match we use scoring function:

   $$score(match(suffix, ast)) = \\ = \sum_{node \in match} \phi(\frac{f(node)}{f(parent(node))}),$$

   where $f(node)$ is the frequency of the matching node and $f(parent(node))$ is it’s parent frequency;

Table 1: Text preprocessing techniques

Name	Description
words	Words
stems	Stems (Porter stemmer in NLTK is used)
coll3	All matches of the taxonomy topic suffixes to the AST built for abstracts. (In this table, the abstracts are split in three word strings.)
coll3.5	Terms from coll3 set consisting of 5 or more characters

3. Then the relevance is estimated by averaging the score of a symbol:

$$relevance(topic, abstract) = SCORE(topic, ast) = \\ = \frac{\sum_{suffix} score(match(suffix, ast))/|suffix|}{|string|},$$

where $|suffix|$ and $|string|$ are the lengths of the suffix and the string.

Note, that "score" is a scaling function, that converts a match score into a relevance estimation. We consider three types of the scaling functions, according to [?], where the AST method was used to categorize e-mails:

- Linear function: $\phi(x) = x$
- Logit function:
$$\phi(x) = \log \frac{x}{1-x} = \log x - \log(1-x)$$
- Root function $\phi(x) = \sqrt{x}$

Of them, only the linear scaling function has an obvious meaning: it stands for the conditional probability of characters averaged over matching fragments (CPAMF).

Big ASTs may suffer from the noise produced by the first levels in the tree. The nodes on these levels have approximately the same frequencies, which are rather high, and equally influence scores of any match. The first level in the AST is responsible for single characters, the second level for pairs of characters and so on. We assume that such short fragments of texts hardly can be meaningful. To check whether these nodes really produce noise in the relevance estimates or not, we compute relevance estimates at which a few initial levels of the AST are not taken into account. We denote by $\phi.X$ such a scaling function ϕ that accounts only those nodes that start from the level $X = 1, 2, ...$

On the whole we use three classes of AST relevance score functions with different scaling functions and noise removal options. The method outputs the list of taxonomy topics in the descending order of their relevance scores.

3. TEXT PREPROCESSING

Vector space model and probabilistic model require a text to be represented as a set of terms. A term is an unmodified word or a word after preprocessing. With AST text model, we use the AST matches to taxonomy topics as terms. These matches may include rather long fragments, quite capable to capture the word order. Table 1 enumerates all preprocessing techniques used in our experiments.

Table 2: Relevance measures used in the experiment

Name	Description
cosine	Cosine relevance measure
BM25	BM25 relevance measure
ast.linear	Linear scaled CPAMF
ast.logit	Logit scaled CPAMF
ast.root	Root scaled CPAMF

4. EXPERIMENTAL SETUP

To computationally compare different relevance measures for annotating research papers one needs: an input dataset, methods for measuring relevance and measures to evaluate computational results Let us describe the experimental setup in more detail.

4.1 Input dataset

We adopt the set of research paper abstracts from the ACM Digital Library. Every abstract is already annotated by the authors with so called index terms that are taxonomy topics from the ACM Computing Classification System 2012. Our task is to obtain relevant taxonomy topics and evaluate how well they match the index terms assigned to the paper. Hence the input dataset is threefold:

1. A collection of 244 abstracts of papers published from January 2007 to March 2013 in the following ACM Journals: ACM Transactions on Knowledge Discovery from Data (TKDD), ACM Transactions on Internet Technology (TOIT) and ACM Transactions on Speech and Language Processing (TSLP)
2. The ACM CCS 2012 taxonomy which comprises 2074 taxonomy topics. There are 6 levels the first of which has 13 major divisions.
3. Index terms that are labels assigned to each of the papers by the author(s).

4.2 Relevance measures

We use the three popular relevance measures defined above to estimate the taxonomy topic-to-text relevance: cosine, BM25 and CPAMF with the three scaling functions defined above.

5. RESULT EVALUATION MEASURES

Index terms are used to check, whether an algorithm has obtained a correct annotation by using this or that relevance measure. We use two popular evaluation measures MAP (Mean Average Precision) and $nDCG$ (normalized Discounted Cumulative Gain) [?]. They are widely used for assessment of sets of ordered (ranged) items that appear, for example, in a recommender system [?] or a news extraction system [?]. Learning to rank [?, ?] appears to be another application of the measures as learning criteria. MAP and $nDCG$ suit our purpose quite well, as using any relevance measure results in an ordered list of taxonomy topics. To apply any of MAP and $nDCG$, several steps are required:

- Taxonomy topics are listed in the descending order of the utilized relevance measure
- First k (top k) taxonomy topics are taken into account disregarding the rest

- The evaluation measure is applied to the top k taxonomy topics

MAP is computed as follows:

$$AveP = \frac{\sum_{k=1}^{n} P(k) \times rel(k)}{|relevant\_topics|}$$

$$MAP = \frac{\sum_{a \in abstracts} AveP(a)}{abstracts},$$

where $P(k)$ is precision at k [11], $rel(k)$ is 1 if kth taxonomy topic is the index term and 0 otherwise, $|relevant\_topics|$, is the number of index terms, n is the number of taxonomy topic under consideration. $AveP$ stands for average precision. This feature is computed for every research paper after which MAP is used to aggregate $AveP$.

$nDCG$ is the proportion of the real discounted cumulative gain (DCG) in the ideal gain DCG: $nDCG_k = \frac{DCG_k}{IDCG_k}$, where $DCG_k = rel(1) + \sum_{i=2}^{k} \frac{rel(i)}{\log_2 i}$ is the number of index terms among top k taxonomy topics, normalized by their rank. $IDCG_k = rel(1) + \sum_{i=2}^{irrelevant\_topics} \frac{1}{\log_2 i}$ is the ideal DCG. Of course $nDCG$ can be averaged [**?**]. Therefore, we compute the average $nDCG$ value for all the 244 abstracts. We take $k = 15$ to be obviously larger than the number of manually assigned index terms.

Also, we propose our own evaluation measure, I_k, which we call $Intersection@k$. For every abstract a we set $i(a) = 1$ if there is an index term among top k taxonomy topics in the corresponding ordering and 0, otherwise. Then $I_k = \sum_{a \in abstracts} i(a)$ This measure has two main advantages. It allows to separate ”good” research papers, that are annotated well by the authors, from those ”complex” ones, that are not; and to set an optimal k for every research paper.

In fact, MAP and $nDCG$ can be also exploited to set the threshold k, but they are not as intuitive as the I_k.

6. RESULTS

There are four measures to evaluate the quality of the ordered lists of taxonomy topics: intersect I_k at k where $k = 5, 15$; MAP and $nDCG$ at $k = 15$. These measures are used to compare both a) relevance measures and b) preprocessing techniques.

Comparing the values of the relevance measures in Table **??**, one can note, that:

- The best results for cosine relevance measure are achieved after using the coll3 preprocessing techniques, since MAP_{15} and $nDCG_{15}$ as well as I_{15} are maximal. However using unmodified words and stems preprocessing techniques leads comparable results.
- For BM25 relevance measure the stemming technique is the winner, as I_5 shows, but at $k = 15$ the unmodified words technique seems to be a better option.
- I_k values provide clear evidence that in general the BM25 relevance measure is not as precise as cosine relevance measure.
- The non-scaling linear CPAMF function provides the highest values according to all three quality measures I_k, MAP_{15} and $nDCG_{15}$.
- The root scaling function achieves a similar accuracy of the labels in contrast to the logit scaling function.

Table 3: Results of using three relevance measures on texts preprocessed with different techniques

Preprocessing technique	I_k		MAP_{15}	$nDCG_{15}$
	I_5	I_{15}		
Cosine relevance measure				
words	44	73	0.0748	0.0245
stems	37	77	0.0788	0.0250
coll3	41	76	0.0911	0.0278
coll3.5	31	71	0.0642	0.0237
BM25 relevance measure				
words	14	52	0.0631	0.0279
stems	21	36	0.0869	0.0259
coll3	15	46	0.0524	0.0224
coll3.5	16	46	0.0577	0.0228
CPAMF relevance measure				
linear.0	75	102	0.3588	0.1124
linear.1	77	105	0.3550	0.1133
linear.2	75	103	0.3486	0.1120
root.0	75	102	0.3657	0.1125
root.1	77	104	0.3561	0.1122
root.2	77	106	0.3497	0.1126
logit.0	36	57	0.1214	0.0450
logit.1	18	33	0.0521	0.0216
logit.2	29	56	0.0780	0.0335

Among the three measures of relevance the linear CPAMF is the most accurate. For example, when CPAMF is used, 7577 index terms are placed among the top five taxonomy topics, whereas the cosine measure and BM25 lead to just 44 and 21 index terms, respectively. The MAP_{15} and $nDCG_{15}$ values for the linear CPAMF are almost four or five times higher than for the cosine measure or BM25. Our hypothesis of the noise coming from the second and third levels of the ASTs is not supported by the experimental results. Clearing only of the very first level, one letter frequencies, is reasonable.

Another way to compare I_k values for different relevance measures is to draw the hit curves, which are piecewise linear curves on the (k, I_k) plane. A hit curve connects I_k values at different k. The higher the hit curve, the more accurate is the relevance measure. If a hit curve for relevance measure A is higher than that for the relevance measure B, one can safely conclude that A is more accurate than B.

The comparison of hit curves on Figure **??** proves that no matter what parameters of relevance measures are, the ast.linear and ast.root relevance measures outperform the cosine and the BM25. Between these two popular relevance measures, cosine and BM25, the former is better than the latter.

7. CONCLUSION

This work explores the issue of computationally annotating/labelling research publications using topics taken from a taxonomy of the research domain under consideration. A specific of this issue is that it concerns mental rather than real objects. Hence the issue of validity of empirical data becomes of great importance. It is not always obvious how to annotate a paper with topics taken from a hierarchical classification system. We suppose that annotations of re-

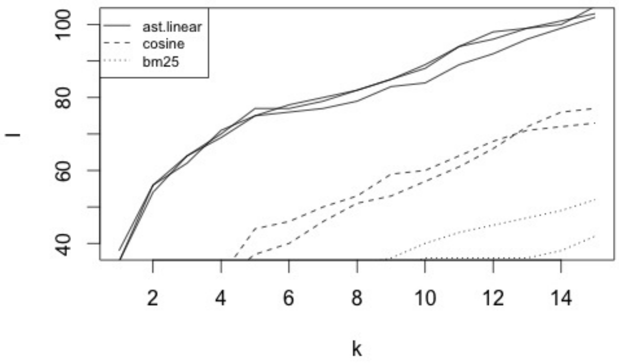

Figure 1: Hit curves of four classes of relevance measures: ast.linear, ast.root, cosine and BM25

search papers made by the authors of research are more adequate than annotations of general documents made by amateurs. Therefore we use research papers published in respected ACM journals and annotated with topics from the ACM CCS 2012. We used string-to-text relevance measure as a main tool for producing self-learning multiple labels. We compared three string-to-text relevance measures: cosine, BM25 and conditional probability of characters averaged over matching fragments in annotated suffix trees (CPAMF).

It appears that the CPAMF outperforms both cosine and BM25 relevance measures in the problem under consideration. Our hypothesis that the first levels of an AST formed by one, two and three letter text fragments give no information for the labelling process proved wrong for the levels 2 and 3, yet correct for the first level. Its removal slightly improves the quality of annotation. Yet in general the match between the automatic annotation and that made by the authors is not quite high, which is not uncommon for self-learning methods in text analysis. Partly this effect is due to differences in the abstracts themselves, but partly this is because the relevance measures take no synonyms into account. Therefore the very first direction to take should be in using synonyms in computing the AST relevance measures. A concept should be represented not by a single string, but by a (fuzzy) set of synonymous topics.

8. REFERENCES

[1] C.D. Manning, P. Raghavan, H. Schutze. *Introduction to information retrieval*, 2008.

[2] F. Sebastiani, Machine learning in automated text categorization. *Journal of ACM Computing Surveys*, 34(1):1-42, 2002.

[3] G. Salton, *Dynamic library and information processing*, 1975.

[4] N. Loukachevitch and B. Dobrov. Large-scale linguistic ontology as a basis for text categorization of legislative documents. *Legal Knowledge and Infromation Systems*, 134:109-134, 2005.

[5] M. Ceci and D. Malerba Classifying web documents in a hierarchy of categories: a comprehensive study. *Journal of Intelligent Information Systems*, 28(1):37-78, 2007.

[6] A.P. Santos and F. Rodrigues. Multi-label hierarchical text vlassification using the ACM taxonomy In *Proceedings of 14th Portuguese Conference on Artificial Intelligence*, pages 553 - 564, Aveiro, Portugal, 2010.

[7] R. Xu, A. Morgan, A. K. Das and A. Garber. Investigation of unsupervised pattern learning techniques for bootstrap construction of a medical treatment lexicon. In *Proceedings of the Workshop on Current Trends in Biomedical Natural Language Processing*, pages 63-70, Stroudsburg, PA, USA, 2009.

[8] J. Grimmer and B. M. Stewart. Text as data: the promise and pittfalls of automatic content analysis methods for political texts. *Political Analysis*, 21(3)267- 297, 2013.

[9] R. Pampapathi, B. Mirkin and M. Levene. A suffix tree approach to anti-spam email filtering. *Machine Learning*, 65(1):309-338, 2008.

[10] B. Mirkin, E. Chernyak and O. Chugunova. Annotated suffix tree method for estimating the string-to-text relevance. *Business-Infromatics*, 21(3):31-41, 2012 (in Russian)

[11] S. Robertson and H. Zaragoza. The probabilistic relevance gramework: BM25 and beyond. *Journal Foundations and Trends in Information Retrieval* 3(4):333-389, 2009

[12] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Information Processing and Management*, 25(5):513-523, 1998.

[13] D. Gusfield. *Algorithms on strings, trees and sequences: computer science and computational biology*, 1997.

[14] I. Cantador, A. Bellogin and D. Vallet. Content-based recommendation in social tagging systems. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 237-240, Barcelona, Spain, 2010.

[15] A. Gupta and P. Kumaraguru. Credibility ranking of tweets during high impact events. In *Proceedings of the first Workshop on Privacy and Security in Online Social Media*, pages 2-8, Lyon, France, 2012.

[16] F. Xia, T. Liu, J. Wang, W. Zhang and H. Li. Listwise approach to learning to rank - theory and algorithm. In *Proceedings of the 25th International Conference on Machine Learning*, pages 1192-1199, Helsinki, Finland, 2008.

[17] K. Duh and K. Kirchhoff. Learning to rank with partially-labeled data. In *Proceedings of the 31st Annual International ACM SIGIR Conference*, pages 251-258, Singapore, 2008.

[18] H. Valizadegan, R. Jin, R. Zhang and J. Mao. Learning to rank by optimizing NDCG measure. *Advances in Neural Information Processing Systems*, 22:1883-1191, 2010.