Vorabfassung des Artikels

Hirschmeier, Stefan, Johannes Melsbach, Detlef Schoder, and Sven Stahlmann. "Improving Recall and Precision in Unsupervised Multi-Label Document Classification Tasks by Combining Word Embeddings with TF-IDF." ECIS 2020 Research Papers, 2020. https://aisel.aisnet.org/ecis2020_rp/99.

IMPROVING RECALL AND PRECISION IN UNSUPERVISED MULTI-LABEL DOCUMENT CLASSIFICATION TASKS BY COMBINING WORD EMBEDDINGS WITH TF-IDF

Abstract

Multi-label document classification is a common task and becomes increasingly important for nowadays business needs. However, generating keywords is not easily done, as next to methodological challenges, labelled training data for classification is not always existent in the desired amount or in the desired quality. Therefore, methods that do not require labelled training data (e.g., unsupervised learning or statistical approaches) are valuable for practice. As none of these approaches alone provides optimal results in terms of recall and precision, we show that it is worth examining existing approaches for complementary strengths in order to combine them. We found such complementary strengths for tf-idf and an unsupervised word embedding method and propose a combined approach. For evaluation, we test the combined approach on a unique dataset of a public broadcaster from Germany and show that precision and recall can be significantly improved.

Keywords: multi-label document classification, unsupervised classification, precision, recall.

1 Introduction

Unsupervised keyword identification for full text documents is a valuable technique for businesses as it allows assigning multiple tags to documents. Businesses in all industries have a growing need to generate metadata for full text sources, e.g., manuscripts in media companies, user-generated content in retailer companies, or patents in research departments.

Often, companies have a fixed set of keywords that need to be assigned to documents[1], typically in magnitude of 1,000 to 10,000. While unsupervised methods usually do not perform as good as supervised learning approaches in classification tasks, they have a significant benefit: no labelled training data is needed. Labelled training data can be a serious problem in practice—the amount of metadata labels is either too small, the quality of the labels is not sufficient, or metadata does not exist at all. Furthermore, as vocabulary changes over time, new labelled training data has to be provided constantly. In most cases, it is not sufficient to stay with a fixed, aging labelled training data set, as new vocabulary is constantly brought into the language (e.g. terms like "Brexit" or "smart speaker"). Unsupervised approaches therefore do not always deliver as good results as supervised classification but are unbeatable from a resource point of view.

The requirements for tag quality can be quite different—it depends on the use case whether results are satisficing (i.e., satisfying and sufficient (Simon, 1959)) or not. In practice, we usually find two scenarios: a) use cases where imperfect data (i.e., inaccurate tags) is acceptable and b) use cases where imperfect data needs to be avoided. The decision whether imperfect data is acceptable is very specific to the business needs (corresponding to the "fitness-for-use" paradigm of information quality (R. Y. Wang and Strong, 1996)). E.g., most companies would agree that offering inaccurate tags for products on a website is not acceptable. For calculating product recommendations, in contrast, inaccurate metadata might be satisficing, as not the tags themselves are presented to consumers, but recommended products, and recommendations cannot be right or wrong from an objective perspective. Imperfect data is also more likely to be acceptable in use cases where humans are assisted by machine and can overtake a corrective part.

Spoken in terms of information retrieval and classification, sometimes a high precision (i.e., no wrong keywords) is favored over a high recall (e.g., when no human intervention is possible) and vice versa depending on the use case. Typically, precision and recall show an inverse relationship and are hard to achieve at the same time, making trade-offs necessary. Classic approaches like the term-frequency–inverse document frequency method (tf-idf) (Hulth, 2003; Baeza-Yates and Ribeiro-Neto, 2010) usually build the baseline for keyword extraction approaches, and more sophisticated approaches compete with this baseline in terms of recall and precision.

It is however sometimes not exhaustive to compare approaches via recall and precision only. Performance measures only show the outer quality of the approaches. Assuming we have a set of keywords {cat, video, cucumber, shock} that correctly describes a document, and approach A comes up with {cat, shock}, while approach B identifies {cat, cucumber}. Both approaches achieve a precision of 50%, but with different predictions. Recall and precision reflect the effectiveness but hide the inner qualities of the approaches. The idea to compare approaches via recall and precision only therefore primarily makes sense for variants of similar approaches. For approaches that follow a very different methodology, also the inner qualities should be compared in order to discover possible opposite strengths and to combine those strengths, if possible.

Interestingly, there is indication that unsupervised document classification with word embeddings have opposite strengths to classic approaches like tf-idf. Therefore, we want to motivate the analysis of strengths and weaknesses of different approaches for unsupervised multi-label text classification. The question is not if one approach outperforms another, but how both approaches can be effectively combined to use the strengths of both and, as an ultimate goal, to improve precision or recall.

[1] Assigned keywords are often called tags, whereas in multi-label text classification, the keywords are typically referred to as classes. We use the terms tagging, classification, annotation or mapping interchangeably throughout the paper.

Therefore, we ask the following question: *How to improve precision or recall for multi-label text classification by combining two approaches on the basis of an analysis of their strengths and weaknesses?*

The remainder of this paper is structured as follows. First, we refer to related work in keyword identification techniques, thereunder classic approaches, approaches with distributed representations, and hybrid approaches. Then, for two approaches, we present an analysis of strengths and weaknesses in Section 3. On this basis, we present our combined unsupervised approach for multi-label document classification in Section 4. In the last section, we provide an evaluation and conclude with a discussion and limitations.

2 Related Work

2.1 Existing Approaches for Keyword Identification in Text Documents without Labelled Training Data

The term frequency–inverse document frequency (tf-idf) (Hulth, 2003; Baeza-Yates and Ribeiro-Neto, 2010) is one of the most fundamental keyword identification methods. tf-idf is a statistical approach that determines the importance of each word in a document. A word is considered important for a document if it has a high frequency in the document but a low frequency in the whole document collection. Therefore, it does not work on a single document. tf-idf is easy to compute but is not designed to match a fixed set of keywords.

Initially designed for automated summarization of texts, the TextRank algorithm can also be used to derive important keywords in a document. It uses graph-based text ranking models that were derived from Google's PageRank algorithm and exists in various variations (Mihalcea and Tarau, 2004; Wan, Yang and Xiao, 2007). TextRank outperforms tf-idf in classic keyword extraction tasks (Wu, Zhang and Ostendorf, 2010). While TextRank is applicable to single documents, it has other shortcomings, as it sometimes leaves out keywords that occur infrequently, but might be important in the context of a document (Zuo, Zhang and Xia, 2017).

Besides tf-idf and TextRank, topic modelling methods such as Latent Dirichlet Allocation (LDA) (Blei, Ng and Jordan, 2003) or Probabilistic Latent Semantic Indexing (PLSI) (Hofmann, 1999) have been developed to extract word collections from documents that best describe the respective content. LDA and PLSI do not rely on manually labelled training data which makes them promising for application on a large scale. Nevertheless, extracted keyword collections have been reported as being hard to interpret (Debortoli, Müller, Junglas and Brocke, 2016) which limits applicability in end-user systems.

The previously mentioned keyword generation approaches make use of the so-called bag-of-words (BoW) representation. Those representations capture the quantitative presence or absence of words of a vocabulary in a document. With only a small portion of words of a vocabulary being present in a given document this leads to very sparse vector representation. Spelling errors, synonyms or word flections further enhance this effect, which makes keyword relevance computations challenging. There are several techniques like stemming, lemmatizing, spelling correction, or synonym mapping that can dampen this effect but only address the symptoms (of sparse vectors), but not the root cause as BoW approaches are limited in capturing semantics (via the context of words).

A promising approach to create semantically meaningful vector representations was introduced by Mikolov et al. (2013). The proposed algorithm, better known as word2vec, creates distributed representations of words and phrases. Other than the BoW approaches it does not rely on word occurrences but is able to create dense vector representations of a word's context. One important characteristic of these word vectors is that semantic similarity is assumed to correspond to arithmetic distance. We consider this representation as a foundation for an unsupervised keyword generation approach. Therefore, related literature will be detailed in the following.

2.2 Classification with Distributed Representations of Documents

The most prominent approach of distributed representation of text documents is the paragraph2vec approach of Le and Mikolov (2014). It proposes two different methods to train local document vectors along with global word vectors. Before Le and Mikolov, other researchers have proposed extensions of the word2vec model to obtain distributed representations of sentences, phrases or documents (Mitchell and Lapata, 2010; Zanzotto, Korkontzelos, Fallucchi and Manandhar, 2010; Grefenstette et al., 2013; Mikolov, Sutskever, et al., 2013). Approaches reach from simple ones that calculate an average of the words in a sentence, phrase or document, to more complex ones, e.g. that combine the word vectors in an order given by a parse tree (Socher, Lin, Ng and Manning, 2011).

Distributed representations of documents on the basis of word2vec approaches allow for a supervised classification of documents with a subsequent classifier, typically a neural network. However, for all subsequent classification tasks on top of word2vec, manual effort is required. Experts need to link a substantial number of documents to classes to build a training set for the classifier.

Companies often face the challenge of having a high number of keywords (in magnitude of 10,000 and more) which represent the classes in a classification task. With the growing number of classes, also a significantly large manually annotated training set is required. Furthermore, this is not a one-time, but ongoing effort. If vocabulary changes over time (as new word like "Brexit" come up), both word2vec model and supervised classifier would have to be trained periodically to reflect newest words.

To sum up, the usage of word embeddings allows to perform a more meaningful feature engineering than solely relying on BoW based approaches. But considering the drawbacks of supervised classification for text documents on top of word embedding approaches, we find that a) a large number of manual tags have to be assigned by experts to obtain a useful labelled training set due to a high number of classes, b) manual tags have to be assigned not only once, but continuously due to changing vocabulary, and c) not only word2vec, but also the subsequent classifier has to be trained periodically. These drawbacks make classifiers on top of word2vec suitable in theory, but less suitable for practice. We therefore propose an approach how to use word embeddings in an unsupervised way in combination with tf-idf.

2.3 Hybrid Classification Approaches

As we propose a combined approach in this paper, we shed light on existing hybrid classification approaches, which might also include supervised approaches. Verma and Arora (2017)propose a hybrid approach for tackling the issue of dynamically matching consumers' queries to the most similar questions available in archives or the web. They compute the average of Word2Vec, tf-idf and tf-idf integrated with Part-Of-Speech-Tagging using cosine similarity. Their evaluation shows that the hybrid approach outperforms other methods applied individually. Arora et al. (2017) use word embeddings, represent sentences by a weighted average of the word vectors, and then modify them using PCA/SVD.

Wang et al. (2016) developed a method for feature extraction of documents by combining Word2Vec and Latent Dirichlet Allocation (LDA). As a result, the generated document vector takes both into account the relationship between documents and topics as well as the relationship among words. The features extracted by their approach lead to an improvement in classification performance.

Yilmaz et al. (2017)introduce a composition of word2vec and k-nearest neighbours (kNN) as a hybrid approach to detect words in colloquial microblogs that represent points of interest (POIs). First, they create word embeddings using word2vec, then they apply the kNN classifier to a specific vector to find the most similar words.

Tae et al. (2006) propose a combination of support vector machine (SVM) and kNN for an automatic text classifier. First, the SVM classifies a document. If the confidence level is too low, kNN will determine a corresponding document class. Experimental results show that this method outperforms SVM applied individually.

Chen et al. (2013) enhance the accuracy of text classification for imbalanced data by combining the advantages of two methods: First, hypersphere-SVM, which has been proven handling imbalanced data, and second, k-congener-nearest-neighbours-SVM, which performs especially well on chaotic imbalanced data.

Kumar and Ravi (2018) propose a hybrid approach for text classification consisting of topic modelling and Class Association Rule Mining (CARM). They test their method based on two publicly available datasets predicting malware. The introduced model showed a high accuracy score.

Nam and Quoc (2015)implemented a hybrid approach for feature selection regarding the classification of documents. By combining a frequency-based with a cluster-based method they manage to utilize the strong aspects of both and manage to outperform other methods.

Tiwari and Singh (2015)develop a feature selection algorithm that leads to an improvement of classification accuracy. They first use a ranking method (RelieF) and choose a subset of k top ranked features that ensure a stable accuracy. Then they create another subset of the original features applying a filtering method (SBS). Finally, testing the union of the two subsets against the features of the methods applied individually results in an enhancement of classification accuracy.

Gao et al. (2008) deal with the topic of web document classification. They show that the combination of a decision tree with a neural network as its categorical value function outperforms single categorization methods.

Glinka et al. (2017) test different combinations of methods for multi-label text classification against each other and discuss them. For the feature selection process, they come up with three hybrid approaches. a) SKB-RFE, a combination of the Select K Best (SKB) method and Recursive Feature Elimination (RFE) method, b) RFE-SKB representing the opposite of the first approach and c) a method that simply eliminates all features that are not identified by SKB and RFE and keeps the rest. An evaluation shows that single SKB and RFE-SKB perform the best regarding classification effectiveness.

2.4 Approaches Examined in this Paper

From all keyword identification approaches, we depict two approaches in detail—the ones, that we analyse in the following section and that subsequently find their way into the combined approach that we propose in Section 4.

Document-tag-cosine-similarity (dt-cs). dt-cs is an unsupervised approach to assign keywords to documents. It uses word2vec (Mikolov, Chen, et al., 2013; Mikolov, Sutskever, et al., 2013) and paragraph2vec (Le and Mikolov, 2014) to train vectors for words and documents of a text corpus. It then transfers trained word vectors to keyword vectors so that both documents and keywords are available in a common vector representation, i.e., both types of vectors have the same structure and dimensionality. As a consequence, it is possible to compute cosine similarities (i.e., standardized dot products) between every pair of document vector and keyword vector. Even though document vectors and keyword vectors were trained with different methods (word2vec for words and paragraph2vec for documents), results show that a high cosine similarity between document vector and keyword vector indeed indicates that the keyword accurately describes the document. This way, the classification task can be simplified to vector operations in a linear algebra. The approach to compare word and document vectors was proposed by (Mohr, 2017).

Term frequency – inverse document frequency (tf-idf). As already introduced, tf-idf is a statistical method that calculates the term frequency (tf) within one document and sets the term frequency in relation to the frequency of the term in all documents (idf) to obtain the term specificity. tf-idf is a pure statistical method and depends on the syntactical occurrences of terms. As such, it cannot process synonyms or any other semantics.

3 Qualitative Differences of Keyword Identification Approaches

In this section we present an analysis of the different qualities of tf-idf and dt-cs, first theoretical, then empirical. For the theoretical analysis, we set up an analysis framework that reflects the qualities of the approaches, containing how the approaches may cope with synonyms, homonyms, false positives, false negatives, and company specific keyword sets.

3.1 Theoretical Analysis

3.1.1 Strengths and weaknesses of tf-idf

tf-idf is known for its easy implementation while still delivering considerable precision and recall. The top keywords that tf-idf provides occur in the document with certainty and in its collectivity generally describe the document well—single keywords might however not be characteristic for the document as they might occur frequently but fail to represent the main topic. This is especially evident when stop words have not been excluded during preprocessing.

Synonyms: tf-idf cannot find synonyms. It even treats conjugations or declinations of the same word differently (therefore, stemming or lemmatizing is useful as preprocessing, otherwise, the same concept will be fragmented among different terms).

Homonyms: tf-idf cannot distinguish homonyms, as it relies on syntactical occurrence only.

False positives: From a pure statistical view, no false positive terms are possible, as each word that tf-idf proposes necessarily appears in the document at least one time. From a classification point of view, however, false positives may exist when a keyword would not characteristically describe a document (i.e., when a human would not choose the keyword as a characteristic tag for the document).

False negatives: tf-idf may well miss out keywords that correctly describe the document, e.g. "politics", as it may happen that the term "politics" does not occur in the document. Also, if a term occurs very rarely, tf-idf would not attribute much statistical value to that term and the term would not be in the set of top n words that describe the document. If it was however an important word, this would be a false negative.

Company specific set of keywords: If a company specific set of keywords should be used for tagging, the easiest way would be to only allow exact matches. This way, however, a quite high amount of tags would be missed out (e.g., tf-idf comes up with "laptop", but the company specific keyword is "notebook"). To come over this, further matching algorithms need to be applied, either with thesauri (map "laptop" to "notebook") or with subsequent classification.

3.1.2 Strengths and weaknesses of dt-cs

The strength of using distributed representations of words and documents clearly lies in the inclusion of context in the representations. This way, representations of words and documents are more robust and even processable in an algorithmic vector space. Classification with word embeddings may therefore come up with words that have never been used in the text explicitly but still describe the text correctly.

Synonyms: As words are represented by their context, dt-cs may well cope with synonyms. Word that share the same context (such as "maracuja" and "passion fruit") are likely to be represented by similar vectors and their cosine similarity is likely to be high.

Homonyms: dt-cs cannot keep homonyms apart.

False positives: dt-cs quite likely produces false positives when words share a similar context. E.g., assuming the words "cats", "dogs", and "pets" share a similar context representation, the keyword "dog" might be erroneously assigned to a text which is purely about cats. The false positive problem of dt-cs could potentially be overcome with a much larger document corpus for training of word2vec. It may however also be the case that our language usage does not allow more specific training and that "cat" and "dog" will always be close vectors, independent of the size of the training data set. Using dt-cs as a tagging approach therefore sometimes resembles a pellet shot—it shoots on several closely related tags, and one or two of them hits.

False negatives: dt-cs may well produce false negatives in a way that relevant words may have a too low cosine similarity with a given document. The problem of false negatives turns out the be connected to the problem of false positives. If "cat" is the word with the highest cosine similarity for a given document about cats being shocked by cucumbers, it is likely that the false positives "dog", "rabbit" and other pets are quite likely to show better cosine similarities with the document than "cucumber" or "shock". This problem may be overcome by applying dt-cs to parts of the documents to reduce the problems of false negatives, but the problem still exists.

Company specific set of keywords: With dt-cs, it is quite easy to match a company-specific set of keywords, as dt-cs is robust with regards to synonyms. As all cosine similarities between documents and possible keywords need to be calculated, the number of possible keywords needs to be finite (in contrast to tf-idf, which may come up with any word that exist in the document—so-called free tags).

Based on our theoretical analysis, we conclude that the characteristics of dt-cs and tf-idf are very opposite.

3.2 Empirical Analysis

For an empirical analysis, we calculated keywords for an example text corpus (data set described in Section 4, the same data set we use for evaluation) by both dt-cs and tf-idf. The results of both approaches are depicted in Figure 1. Additionally, manually assigned tags are provided as a reference.

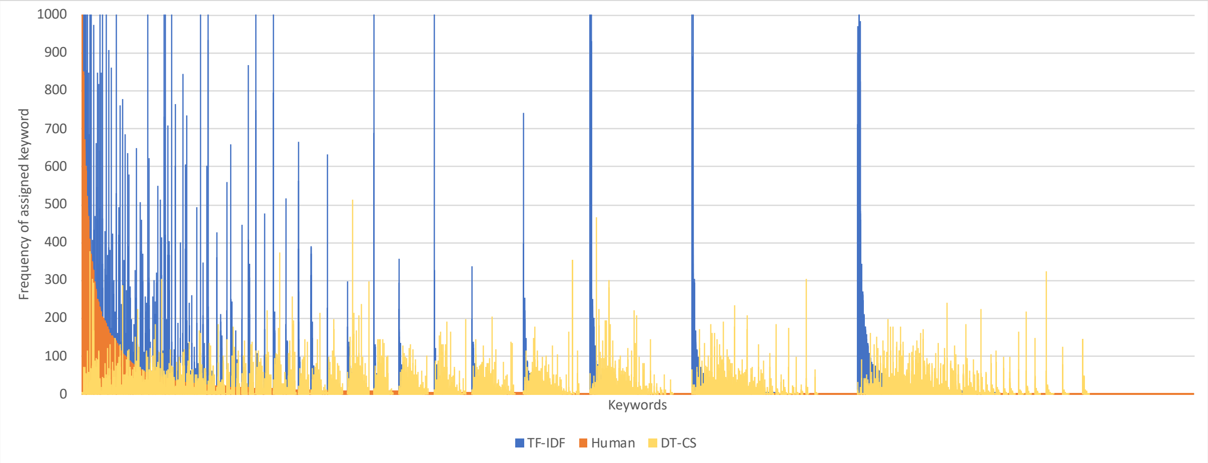

Figure 1: Comparison of frequency of assigned tags: a) manually assigned tags (orange), b) tf-idf (blue), and c) dt-cs (yellow)

Figure 1 shows approximately 10,000 keywords on the x-axis and their frequency on the y-axis. The names of the keywords on the x-axis are not shown (as an example, the three most frequent keywords to the left of the graph are "portrait", "scientist" and "IT"). Figure 1 shows that the manual assignment of tags by humans (orange) is not well balanced among the data set. The most frequent keywords form the long neck to the left of the orange graph. Rarely assigned keywords from the long tail (mostly hidden under the yellow graph). In comparison, the distribution of tags identified by tf-idf is shown in blue (behind the orange graph). The distribution of the tags identified by dt-cs is depicted in yellow. The x-axis is sorted after the most frequent manually assigned tags (first order, orange), the most frequent tf-idf tags (second order, blue) and the most frequent dt-cs tags (third order, yellow). The first, second and third order is the reason why the characteristic power-law distribution shape reappears several times in blue and yellow.

The graph is not intended to display which approach is more correct than the other, as the manually assigned tags are surely correct, but tagging was not necessarily exhaustively done. Humans naturally have a limited cognitive availability of all 10,000 keywords and tend to reuse the same keywords. Also, humans are not always perfectly rational in their decisions—time pressure, extensive knowledge of the keyword set, experience, fatigue and daily condition may influence the information processing and, as a result, the quality of the work. This may partly explain why the orange graph is skewed as well.

Figure 1 clearly shows that the computational approaches tf-idf (blue) and dt-cs (yellow) behave quite differently. While the tag distribution of both approaches does not resemble very much the manually assigned tag distribution (orange), the dt-cs tags (yellow) seem less skewed than the tf-idf tags (blue). In fact, the 10% most frequent keywords identified by tf-idf account for 77% of all tf-idf tags, while the 10% most frequent keywords identified by dt-cs account for 59% of all dt-cs tags. Obviously, dt-cs tags more diverse than tf-idf does. Seen from a macro perspective, both approaches identify different keywords, which is another indication for different qualities of the approaches.

4 A Combined Unsupervised Approach for Text Classification

Given a finite set of keywords that should be assigned to unlabeled documents, methodologically spoken, we have a classification task to fulfil. The term "classification" commonly refers to a supervised task, where a model is trained on labelled training data and then allows to classify new documents according to the trained model. Here, we use the term "classification" also for unsupervised methods that allow to assign tags to documents. Our approach requires the following resources (only):

uDocs. A set of unlabeled documents that need to be tagged

finTags. A finite set of predefined tags (keywords /classes) that should be assigned to the *uDocs*

No labelled training data is needed. The approach we depict consists of four steps, which we describe in the following.

4.1 Approach Details

First step (representation): Calculate tf-idf representations for all words of the *uDocs*. Also establish distributed representations for words and documents by using the Paragraph Vector Distributed Memory (PV-DM) approach of Le and Mikolov (2014).

Second step (filter): Filter out all tf-idf word representations that are not included in the set of *finTags*. Also, filter out all dt-cs word representations that are not included in the set of *finTags*. Keep however all vector representations (distributed representations) of documents.

Third step (top n): Identify the top n keywords of both approaches. For tf-idf, sort the list of keywords after tf-idf scores and cut off after n. To identify top n keywords of dt-cs, compute cosine similarities (via dot products) as distances between documents vectors and word vectors to find best matches. For each document, we compute the cosine similarities to all word vectors and choose the top n ones with the highest cosine similarity (Figure 2). In the end, we obtain both top n matches of tf-idf and dt-cs.

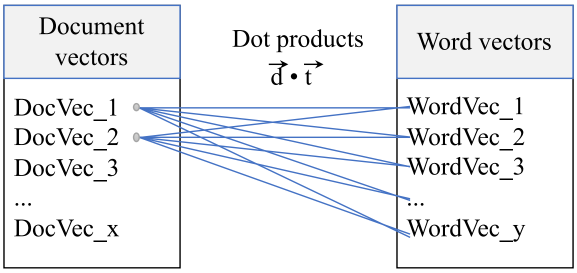

Figure 2: Calculation of dot products with all word vectors to find top cosine similarities for each document

Fourth step (intersect or unite): Build the intersection or union of both approaches' top n outcome. To obtain high precision, use the intersection of the top n keywords of both approaches. To obtain a richer recall, use the union of the top n keywords of both approaches.

The steps of the combined approach are depicted in Figure 3. We do not present text preprocessing as part of the approach. Of course, adequate text preprocessing should be made.

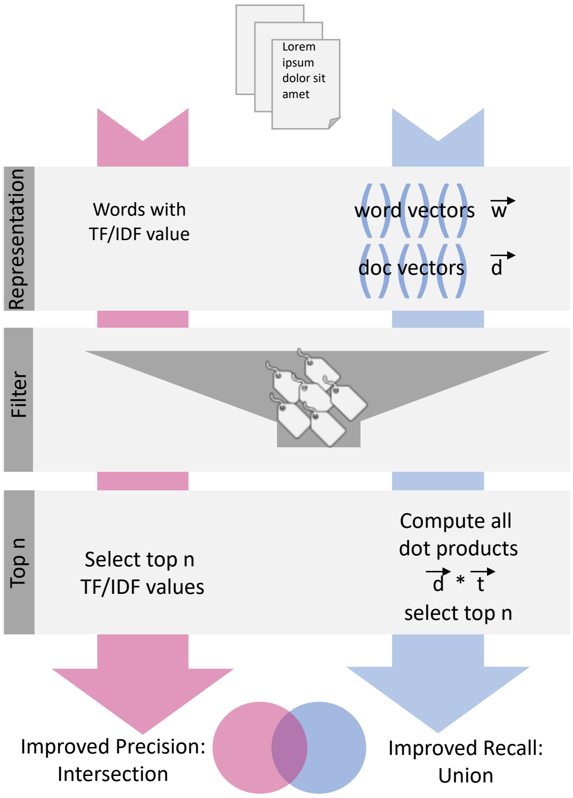

Figure 3: Combined approach for unsupervised multi-label text classification with tf-idf (left) and dt-cs (right)

4.2 Advantages of a Combined Unsupervised Approach

The strengths of dt-cs and tf-idf can be combined for two purposes: to achieve a higher precision or a richer recall.

Combining the strengths for high precision. dt-cs follows a fundamentally different approach than tf-idf. So, when both approaches predict the same keyword, the probability increases that this keyword is a true positive. As a result, the precision of the intersection of both approaches is supposed to increase.

Combining the strengths for a richer recall. As tf-idf cannot predict keywords that syntactically do not appear in the text, the recall of tf-idf is limited to exact occurrences of keywords in the text. In contrast, dt-cs is not limited to that. In combination, the approaches offer the possibility to obtain a much richer recall than one approach alone can achieve.

5 Evaluation

5.1 Evaluation Dataset and Metrics

The data set we use for evaluation covers 63,165 manuscripts from a nation-wide public radio broadcaster from Germany with about 70 million words in total. Each document has a minimum length of 100 words and is written in German. The broadcaster has an archive process where archivists manually assign keywords to the manuscripts. For all 63,165 documents, these manually annotated keywords are provided. The set of keywords consist of 9,885 keywords[2], of which the archivists however only used 7,158 in the 63,165 documents.

[2] In fact, 2351 more keywords exist on higher levels of a taxonomy. We however only regard the leaves of the taxonomy in this paper.

5.2 Evaluation Implementation and Results

For tf-idf, we used the implementation of scikit-learn[3], and for word embeddings, we used the gensim library[4]. As text preprocessing for dt-cs, we replaced all capital letters to small letters, all numbers by their word equivalents (“4” to “four”), replaced all special characters and eliminated all punctuation, as is common for word2vec preprocessing. Also, we concatenated all n-grams keywords with underscore and replaced those n-gams in all documents accordingly. After this, we eliminated stop words.

For each document vector of *uDocs*, we calculated the cosine similarities with all word vectors of *finTags*, resulting in 63,165 x 9,885 dot products. The best cosine similarity for each document represents the best prediction for the document.

Both approaches, tf-idf and dt-cs, were only allowed to predict keywords that exist in *finTags*. After following the four steps of our approach, we noticed that values for recall and precision are generally rather low for both tf-idf and dt-cs (Figure 4). This is due to our specific data set: only few tags were assigned by archivists, so the tags provided by humans are not comprehensive (see Section 3.2). As a result, values of recall and precision of both approaches appear assumingly lower than necessary. For our dataset, tf-idf performs better than dt-cs (with however hyperparameters not optimized).

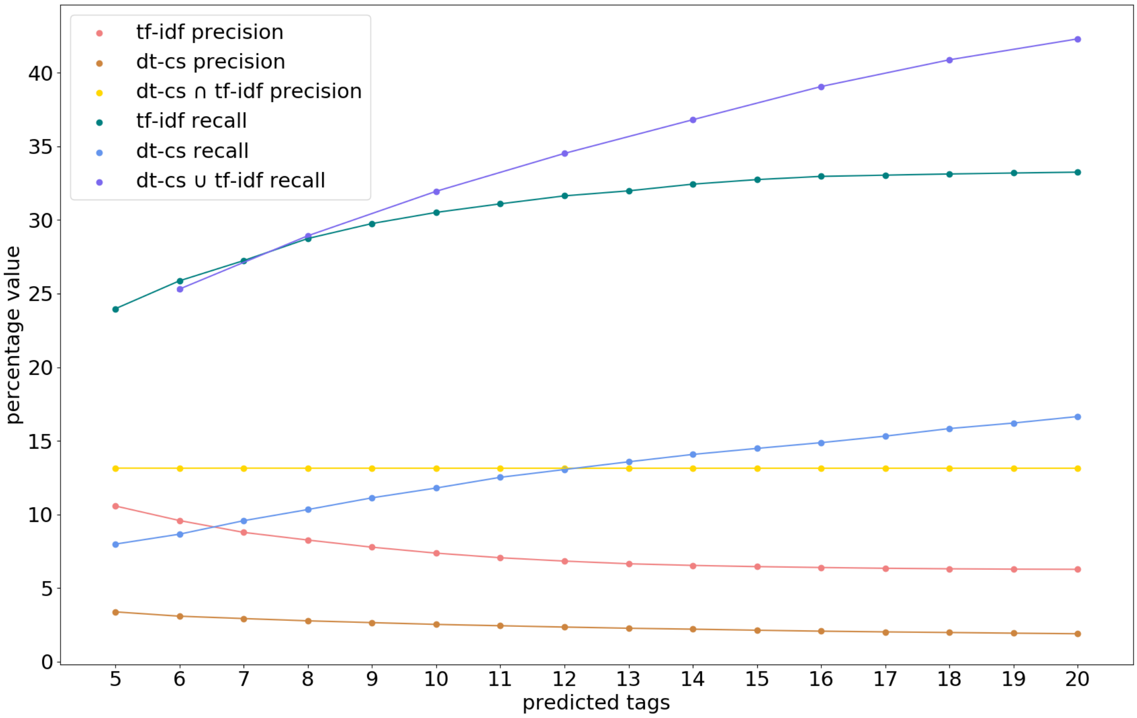

Figure 4: *Recall and precision for tf-idf and dt-cs, the precision of the intersection tf-idf ∩ dt-cs and the union of the recall tf-idf ∪ dt-cs (depending on the number of predicted tags)*

Precision. Values for tf-idf precision (red, over 10% and slightly decreasing) are better than for dt-cs precision (orange, almost 4% and decreasing). Interestingly, the intersection of both (yellow) makes the precision rise significantly, even higher than the sum of both, to a stable level at 13.1%. The yellow graph looks like a straight line, but in fact, the values differ slightly[5].

It has however to be noted that in almost half of all cases, no keyword could be predicted at all by tf-idf. In these cases, the precision is n/a per definition (because of division by zero). If we include these cases and treat them with a precision of 0 instead of n/a, the resulting average precision is 13.1% (as

[3] http://scikit-learn.org

[4] https://radimrehurek.com/gensim

[5] The exact values are 13.156%, 13.140%, 13.144%, 13.1408%, 13.141%, 13.138%, 13.137%, 13.138% etc.

depicted in the graph). If we however exclude these cases, we achieve a precision of 24.5%, almost the double value of 13.1%. Independent of numbers, the qualitative insight is that the intersection improves precision dramatically compared to the best precision of one single approach.

Recall. The recall of tf-idf (green) and dt-cs (blue) increases steadily with the number of predicted tags. Analogously to precision, the recall can be improved by building the union set of both approaches' predictions (purple). By doing so, the value for recall could be improved up to 40%. Compared to the tf-idf recall values, the higher recall proves that the dt-cs approach is able to find further correct keywords that have not been identified by tf-idf (as word embeddings manage to grasp the context of words).

Note that in the union recall (purple line) only even numbers have been calculated. This is because we take 50% of keywords from tf-idf and 50% from dt-cs, so odd numbers are impossible.

	5	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**
a	10.6%	9.6%	8.8%	8.3%	7.8%	7.4%	7.1%	6.8%	6.7%	6.5%	6.5%	6.4%	6.4%	6.3%	6.3%
b	24.0%	25.9%	27.2%	28.7%	29.8%	30.5%	31.1%	31.6%	32.0%	32.4%	32.7%	33.0%	33.1%	33.1%	33.3%
c	3.4%	3.1%	2.9%	2.8%	2.7%	2.5%	2.4%	2.4%	2.3%	2.2%	2.1%	2.1%	2.0%	2.0%	1.9%
d	8.0%	8.7%	9.6%	10.3%	11.1%	11.8%	12.5%	13.1%	13.6%	14.1%	14.5%	14.9%	15.3%	15.8%	16.2%
e	13.2%	13.1%	13.1%	13.1%	13.1%	13.1%	13.1%	13.1%	13.1%	13.1%	13.1%	13.1%	13.1%	13.1%	13.1%
f		25.3%		28.9%		32.0%		34.5%		36.8%		39.1%		40.9%	

a: tf-idf precision, b: tf-idf recall, c: dt-cs precision, d: dt-cs recall, e: tf-idf ∩ dt-cs precision, f: tf-idf ∪ dt-cs recall

Table 1: Values of Figure 4

6 Discussion, Limitations and Further Research

In our research, we contributed an unsupervised approach for multi-label document classification by combining two approaches with complementary strengths and showed that this combined approach can significantly boost recall and precision. The precision we obtained was even higher than the sum of precisions of both approaches alone. We also gave an insight into complementary strengths of dt-cs and tf-idf and motivated that it may be valuable to analyse existing approaches for their characteristics and find other fruitful combinations of approaches.

In our evaluation, all values of precision and recall of our evaluation seem generally low. This may be due to the fact that 9,885 is a high number of classes and archivists did not assign all keywords that would have been useful. It might also depend on the uniqueness of our dataset and unknown rules that archivists may follow when assigning tags to documents. In this regard, our evaluation data set may be imperfect, which affects our values of precision and recall.

Our dataset has certain specifics that makes both approaches look poor. As values of precision and recall highly depend on the dataset, these numbers cannot be compared to values achieved on other datasets. Still, the dataset we used has its justification, as it is a real-world dataset that reflects specifics of datasets in practice. We could show that precision can be boosted by intersecting the top n predictions of both approaches. Similarly, recall can be improved by taking the union set of predictions, in excess to the recall one single approach alone can deliver.

Discussing the relevance of a combined approach for practice, we refer back to the two scenarios given in the introduction: use cases with imperfect data allowed and use cases with imperfect data unfavoured.

Full automation often requires high degrees of certainty. When identified keywords should be automatically pushed to a productive system for navigation, browsing or just for display (e.g. a company's website), without further involvement of humans, high values of precision are usually required by companies. In such cases, few tags with high precision are favorable in contrast to more tags with lower precision. It might even be better not to predict any tag than an incorrect one. It is questionable if the precision we obtained in our evaluation is good enough to serve use cases with need for perfect data. On other datasets and with tuned parameters however, this value might be higher and satisficing for some use cases.

Decision support tasks, in contrast, foresee machines assisting humans, with humans in the role of being a corrective to machines errors. In the case of document classification, machine's task consists of proposing keywords and human's task of evaluating them. For humans, it is significantly easier to select or deselect keywords out of a set of, e.g., 20 that have been preliminary identified by machine, than to make up keywords on their own from a set of several thousand keywords (9,885 in our case). For decision support on keyword identification, a rich recall is favourable. The richer the recall, the richer the results of the human (assuming that humans only select or deselect keywords proposed by machine).

Our combined approach also has some limitations. First, precision and recall can only be improved by two separate mechanisms (intersection and union). Second, as both approaches cannot cope with homonyms, we inherit this weakness.

Considering further research, we still need to tweak our approach by optimizing the hyperparameters for word and document embeddings, and also test it on other datasets that are publicly available. This way, we will be able to make our achieved results better comparable to other methods and point out even more clearly the benefit of the combined approach.

Also, we want to motivate to search for other combinations of methods that have complementary strengths. While tf-idf was used in this paper, it is imaginable that a more sophisticated approach, or a combination of three approaches might even further boost recall and precision.

References

Arora, S., Y. Liang and T. Ma. (2017). "A Simple but Tough-to-Beat Baseline for Sentence Embeddings." Presented at the ICLR.

Baeza-Yates, R. and B. Ribeiro-Neto. (2010). *Modern Information Retrieval.* New York: Addison Wesley.

Blei, D. M., A. Y. Ng and M. I. Jordan. (2003). "Latent Dirichlet Allocation." *The Journal of Machine Learning Research*, *3*, 993–1022.

Chen, Y. H., Y. F. Zheng, J. F. Pan and N. Yang. (2013). "A Hybrid Text Classification Method Based on K-Congener-Nearest-Neighbors and Hypersphere Support Vector Machine." In: *2013 International Conference on Information Technology and Applications* (pp. 493–497). Chengdu, China: IEEE.

Debortoli, S., O. Müller, I. Junglas and J. vom Brocke. (2016). "Text Mining For Information Systems Researchers: An Annotated Topic Modeling Tutorial." *Communications of the Association for Information Systems*, *39*(1).

Gao, H., Y. Fu and J.-P. Li. (2008). "Classification of Sensitive Web Documents."

Glinka, K., R. Wozniak and D. Zakrzewska. (2017). "Improving Multi-label Medical Text Classification by Feature Selection." In: *2017 IEEE 26th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE)* (pp. 176–181). Poznan, Poland: IEEE.

Grefenstette, E., G. Dinu, Y.-Z. Zhang, M. Sadrzadeh and M. Baroni. (2013). "Multi-Step Regression Learning for Compositional Distributional Semantics." Presented at the International Conference on Computational Semantics (IWCS).

Hofmann, T. (1999). "Probabilistic Latent Semantic Analysis." In: *Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence* (pp. 289–296). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Hulth, A. (2003). "Improved Automatic Keyword Extraction Given More Linguistic Knowledge." In: *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing* (pp. 216–223). Stroudsburg, PA, USA: Association for Computational Linguistics.

Kumar, B. S. and V. Ravi. (2018). "A Hybrid Approach Using Topic Modeling and Class-Association Rule Mining for Text Classification: the Case of Malware Detection." In: *2018 IEEE 17th International Conference on Cognitive Informatics & Cognitive Computing (ICCI*CC)* (pp. 261–268). Berkeley, CA: IEEE.

Le, Q. and T. Mikolov. (2014). "Distributed Representations of Sentences and Documents." In: *International Conference on Machine Learning* (pp. 1188–1196).

Mihalcea, R. and P. Tarau. (2004). "TextRank: Bringing Order into Text" (pp. 404–411). Presented at the Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing.

Mikolov, T., K. Chen, G. Corrado and J. Dean. (2013). "Efficient Estimation of Word Representations in Vector Space." *ICLR Workshop*.

Mikolov, T., I. Sutskever, K. Chen, G. Corrado and J. Dean. (2013). "Distributed Representations of Words and Phrases and Their Compositionality." In: *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2* (pp. 3111–3119). USA: Curran Associates Inc.

Mitchell, J. and M. Lapata. (2010). "Composition in Distributional Models of Semantics." *Cognitive Science*, *34*(8), 1388–1429.

Mohr, G. (2017, July 14). "Doc2Vec - How to get similarity between word and doc vectors?"

Nam, L. N. H. and H. B. Quoc. (2015). "A Combined Approach for Filter Feature Selection in Document Classification." In: *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)* (pp. 317–324). Vietri sul Mare, Italy: IEEE.

Simon, H. A. (1959). "Theories of Decision-Making in Economics and Behavioral Science." *The American Economic Review*, *49*(3), 253–283.

Socher, R., C. C.-Y. Lin, A. Y. Ng and C. D. Manning. (2011). "Parsing Natural Scenes and Natural Language with Recursive Neural Networks." In: *International Conference on Machine Learning* (pp. 129–136). Omnipress.

Tae, Y.-S., J. Son, M. Kong, J.-S. Lee, S.-B. Park and S.-J. Lee. (2006). *A Hybrid Approach to Error Reduction of Support Vector Machines in Document Classification* (Vol. 2006).

Tiwari, S. and B. Singh. (2015). "A hybrid approach for feature selection." In: *2015 Third International Conference on Image Information Processing (ICIIP)* (pp. 277–280). Waknaghat, India: IEEE.

Verma, A. and A. Arora. (2017). "Reflexive hybrid approach to provide precise answer of user desired frequently asked question." In: *2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence* (pp. 159–163). Noida, India: IEEE.

Wan, X., J. Yang and J. Xiao. (2007). "Towards an Iterative Reinforcement Approach for Simultaneous Document Summarization and Keyword Extraction." In: *45th Annual Meeting of the Association of Computational Linguistics*.

Wang, R. Y. and D. M. Strong. (1996). "Beyond Accuracy: What Data Quality Means to data Consumers." *Journal of Management Information Systems*, *12*(4), 5–33.

Wang, Z., L. Ma and Y. Zhang. (2016). "A Hybrid Document Feature Extraction Method Using Latent Dirichlet Allocation and Word2Vec." In: *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)* (pp. 98–103). Changsha, China: IEEE.

Wu, W., B. Zhang and M. Ostendorf. (2010). "Automatic Generation of Personalized Annotation Tags for Twitter Users." In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics* (pp. 689–692). Stroudsburg, PA, USA: Association for Computational Linguistics.

Yilmaz, T., P. Karagoz and Y. Kavurucu. (2017). "Exploring what makes it a POI." In: *2017 IEEE SmartWorld, Ubiquitous Intelligence & Computing, Advanced & Trusted Computed, Scalable Computing & Communications, Cloud & Big Data Computing, Internet of People and Smart City Innovation (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)* (pp. 1–6). San Francisco, CA: IEEE.

Zanzotto, F. M., I. Korkontzelos, F. Fallucchi and S. Manandhar. (2010). "Estimating Linear Models for Compositional Distributional Semantics." In: *Proceedings of the 23rd International Conference on Computational Linguistics* (pp. 1263–1271). Stroudsburg, PA, USA: Association for Computational Linguistics.

Zuo, X., S. Zhang and J. Xia. (2017). "The enhancement of TextRank algorithm by using word2vec and its application on topic extraction." *Journal of Physics: Conference Series, 887*.