Vorabfassung des Artikels

Johannes Melsbach, Detlef Schoder, and Sven Stahlmann. "Unsupervised Multi-Label Document Classification for Large Taxonomies Using Word Embeddings," 2019.

Unsupervised Multi-Label Document Classification for Large Taxonomies Using Word Embeddings

Abstract

More and more businesses are in need for metadata for their documents. However, automatic generation for metadata is not easy, as for supervised document classification, a significant amount of labelled training data is needed, which is not always present in the desired amount or quality. Often, documents need to be tagged with a predefined set of company specific keyword that are organized in a taxonomy. We present an unsupervised approach to perform multi-label document classification for large taxonomies using word embeddings and evaluate it with a dataset of a public broadcaster under two assumptions – "perfect rationality" and "bounded rationality", which allows the approach to outperform the average performance humans. We point out strengths of the approach compared to syntactical approaches like tf-idf and show that also allowing to match keywords on a higher taxonomy level can significantly increase precision and recall.

1. Introduction

More and more businesses are in need for metadata for their documents. Metadata is required to get users engaged with products and services on the internet, e.g. to provide navigation tags during product search in web shops, to set up content-based recommender systems, to provide short summaries, or to optimize content for search engines. At the same time, more and more full text data is generated, like social media content, product descriptions, and other text sources. Regardless for which application scenario metadata is needed, it is challenging for companies to obtain it. Many businesses have to find or reconsider their solution how to generate metadata, either manually or automatically. Both ways have their shortcomings.

In manual metadata generation–especially in data rich applications–the human factor becomes a bottleneck. When humans have to assign tags[1] to documents, they often have restrictions and limited resources: time-pressure and their own imaginativeness, experience and condition determine quality and quantity of the output. The result is that usually only few tags are assigned, not all documents are covered, or keyword generation is subjective and not repeatable. As an alternative, automatic keyword generation techniques can provide higher quantity, but have other drawbacks, as the keyword generation techniques are often either not accurate enough and produce too many keywords that are irrelevant, too specific or too general, or do not match the vocabulary used by the organization. In other words, the information quality of automatically generated keywords in the comprehension of fitness-for-use [1] is hard to obtain and therefore often limited.

Often, the vocabulary of the organization is organized in a hierarchical form, in a taxonomy. If the number of terms in a company's taxonomy is small enough and labelled training data is at hand, classifiers can be trained. In quite some contexts, however, numerous classes exist within the company context, e.g. more than 10,000. In those cases, a huge labelled training set is required. Assuming 5,000 labelled training instances per class would lead to a good classification, the labelled training set needs to be in the magnitude of 10 million document-class relations. Assuming further, that a typical labelled training set would not have an equal distribution among classes, the

[1] The manual process of assigning keywords to a document is often called tagging, whereas in multi-label text classification, the keywords are typically called classes. We use the terms tagging, classification, annotation or mapping interchangeably throughout the paper.

training set needs to be even larger. But not all companies have huge labelled training sets at hand. Often, the opposite is the reality: Training data is only available to a limited extent or is not present at all.

Another factor makes supervised classification difficult: temporal dynamics. Over time, both the vocabulary of the documents and the vocabulary of the taxonomies change. For new upcoming words (like "Brexit" or "smart speaker"), a formerly trained classifier might perform poor on accuracy, and each time vocabulary needs to be adapted, classifiers have to be trained again. The problem of having enough labelled training data at hand is therefore a constant companion and comes again and again in new facets.

Our research we document in this paper can be best described by asking the question: *How can multi-label classification of documents be done for large taxonomies (when typically not enough labelled training data is at hand for supervised classification)?*

2. Related Work

2.1. Existing Approaches for Unsupervised Keyword Identification in Text Documents

One of the most fundamental approaches in keyword identification is the term-frequency/inverse document frequency method (tf-idf) [2,3]. tf-idf identifies keywords that quantitatively best differentiate documents within a document collection, is unsupervised and easy to compute. Unfortunately, tf-idf does not allow to match keywords to items of an existing taxonomy. Furthermore, it is not applicable to single documents, as the measure requires a document collection to evaluate the keywords' descriptiveness.

TextRank is an algorithm that uses graph-based text ranking models that were derived from Google's PageRank algorithm and exists in various variations [4,5]. TextRank outperforms tf-idf in classic keyword extraction tasks [6]. TextRank however also has shortcomings, as it sometimes leaves out keywords that occur rarely, but are important in the context of the document [7]. Next to tf-idf and TextRank, topic modelling methods such as PLSI [8] or LDA [9] have been proposed to identify word collections from documents that aim to describe topics the documents deal with. Similar to tf-idf, topic modelling methods are unsupervised which makes them promising for application on a large scale. Nevertheless, extracted keyword collections have been reported as being hard to interpret [10] which limits applicability in end-user systems. Furthermore, additional supervised processing is necessary to match extracted keywords to existing taxonomies. In this case, the results of tf-idf-based, TextRank-based, or topic modelling methods have to be used as features in supervised document classification, which leads to challenges in precision and recall when the predefined taxonomy is large, or taxonomy items are not perfectly distinctive.

Previously proposed keyword generation approaches all rely on a so-called bag-of-words (BoW) perspective. This means, that the quantitative presence or absence of words is the foundation of any representation vector (e.g. document or word). In keyword relevance computations this is challenging, as spelling errors, synonyms or word flections lead to large but sparse matrices. However, applying methods like stemming, lemmatizing, spelling correction, or synonym mapping does only address the symptoms (of sparse matrices), but not the root cause as BoW approaches are limited in capturing semantics like context. Often, low accuracy is the result of mapping an arbitrary set of keywords to a predefined taxonomy.

A promising approach to address BoW context loss was proposed by Mikolov et al. [11]. Word2Vec creates a distributed representation of words[2] (or larger entities such as phrases or documents) that does not depend on the presence or absence of the target word but creates a vector representation of a word's context. One important characteristic of these word embeddings is that semantic similarity corresponds to arithmetic distance. The paper at hand considers this representation as a foundation for unsupervised keyword generation approaches. Therefore, related literature will be detailed in the following.

2.1. Classification with Distributed Represen-tations of Documents

The most prominent approach of distributed representation of text documents is the paragraph2vec approach of Le and Mikolov [12]. 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 [13–16]. Approaches reach from simple ones that

[2] Distributed representation is often connected to the term "word embeddings". Both denote the output of the word2vec approach by Mikolov et al. [11].

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 [17].

Distributed representations of documents on the basis of word2vec approaches allow for a 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 documents to classes to form a training set for the classifier.

For taxonomies, we typically face the challenge of having a high number of keywords (in magnitude of 10,000 to 100,000) and therefore as many classes for the 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 effort, but ongoing. If vocabulary changes over time (as new word like "Brexit" come up), both word2vec model and classifier 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 training set due to a high number of classes, b) manual tags have to be assigned not only initially, 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 in practice.

3. An Unsupervised Approach to Generate Tags for Large Taxonomies

3.1. Approach Requirements

Founded in the need for metadata of an example company, a public broadcaster in Germany, we elicited three requirements for keyword annotation, which we depict in short:

R1. Keyword annotation should take place automatically with a minimum of human effort.
R2. Identified keywords should match an organization's specific taxonomy (in the magnitude of >10,000).
R3. New upcoming words (like "Brexit" or "smart speaker") in documents should be matched as well, taking temporal dynamics of the vocabulary into account.

In the following, we depict an approach to match the elicited requirements.

3.2. Approach Details

Assigning a definite set of taxonomy keywords to documents is a classification task. Given a document space and a taxonomy without any relation between them that could be used for supervised learning, we present an approach that does not require manually annotated training data and is robust against changes in vocabulary. Our approach consists of three steps:

First step: Train word vectors and document vectors on a company specific data set which we denote as world corpus vectors (WCV). Here we use the Paragraph Vector Distributed Memory (PV-DM) approach of Le and Mikolov [12].

Second step: Under the prerequisite, that the keywords of the predefined taxonomy are also found within the world corpus (see first step), we collect the word vectors generated for the keywords in the taxonomy. As a result, we receive a common vector representation of both documents and taxonomy keywords.

Third step: We compute dot products as distances between documents vectors and word vectors to find best matches. For each document, we compute the dot products to all taxonomy word vectors and choose the ones with the highest cosine similarity (Figure 1). Multiple taxonomy keywords can be assigned by choosing the top n keywords or keywords over a certain threshold of similarity or both. Emerging keywords (e.g. due to changing vocabulary or emerging topics) might be discovered when word vectors from the WCV are found similar to a document vector, but do not exist in the predefined taxonomy.

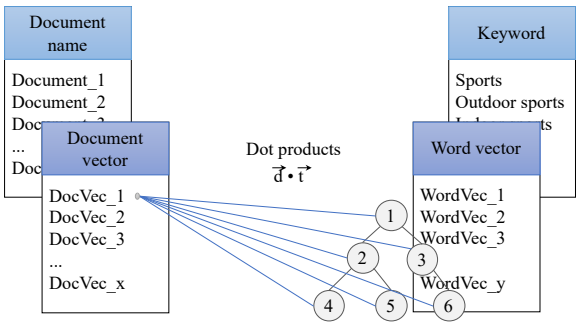

Figure 1. Calculation of dot products for each document vector with all word vectors of the taxonomy

This way, we can simplify the problem of keyword classification and discovery to algebraic vector operations. The approach is not truly unsupervised, as some authors emphasize that word2vec is not unsupervised, but self-supervised, as some error backpropagation takes place through correct and incorrect predictions [18]. But in the sense that no annotation of human experts is required for training, the method can be considered as unsupervised.

Figure 2 depicts the approach in contrast to simple classification approaches (Figure 2 left), where the classification is done on single words or n-gram representations only. Figure 2 (middle) shows advanced classification approaches, that use distributed representations as a better input for the classification task. Figure 2 (right) depicts the approach presented here that classifies via cosine similarity after transforming the taxonomy into vector representation as well. Considering the outlined requirements, R1 and R3 distinguish the approach from classifiers as depicted in Figure 2 (left and middle), and R2 distinguishes the approach from tf-idf and other unsupervised approaches that do not use word embeddings.

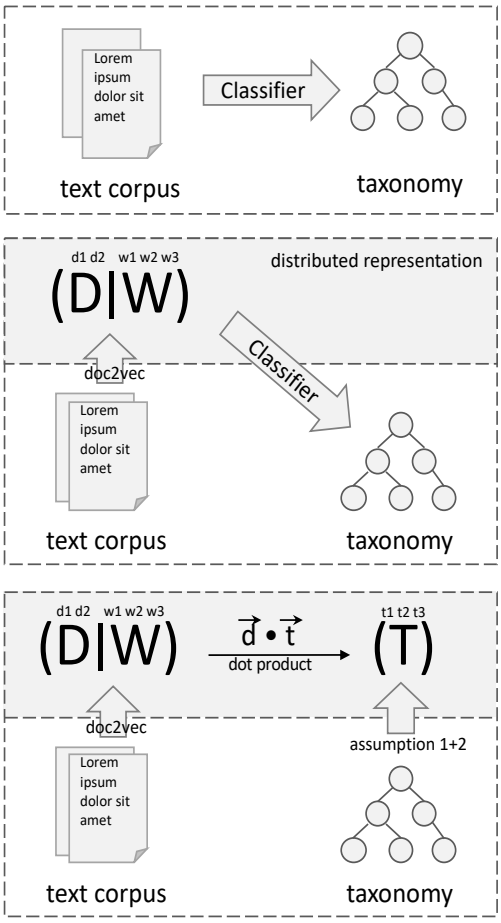

Figure 2. (top) simple classification without distributed representation, (middle) advanced classification with distributed representation of words and documents, (bottom) our proposed approach

3.3. Approach Assumptions

The feasibility and simplicity of the approach bases on three assumptions we made.

Assumption 1: We assume that it is valid to make a lookup for keyword word vectors.

The keywords in the taxonomy have no context except from their position in the taxonomy, which is a rather weak information compared to the richness of word vectors computed by word2vec. As we have no possibility to compute a rich context of single taxonomy keywords, we assume it is valid to make a lookup on the word vectors of the WCV and to use these vectors for the keywords in the taxonomy as well.

Assumption 2: The keywords in the taxonomy are part of the vocabulary of the document corpus.

When new words come up in the document corpus, the taxonomy must either be updated as well, or the taxonomy is required to consist of a more general, slower changing vocabulary that does not adopt fashion words. In general, the assumption that the vocabulary of the taxonomy is a subset of the vocabulary of the document corpus seems valid, as the taxonomy is designed to describe the documents. In other words, no case should exist where a taxonomy keyword is not reflected in the document corpus, otherwise the taxonomy would fail to reflect the document corpus.

Assumption 3: It is valid to compare word vectors with document vectors.

Word vectors and documents vectors (the output of the trained neural network) are of the same dimensionality and structure. Therefore, technically, it is possible to calculate dot products between any pair of vectors. However, it needs to be discussed if document vectors and word vectors can be compared in a semantic way. Word vectors represent the context of a word, which is the set of words typically surrounding the focal word. Document vectors however represent the context of a document which is less easy to imagine. In a way, the training procedure is similar, but the training input is a different one, so one could argue that there is a systematic difference between word vectors and document vectors. Lau and Baldwin [19] state that the qualitative difference between word vectors and document vectors remains unclear and try to give an impression of the difference with an example document. Apart from that, the comparability of word vectors and document vectors has not been thoroughly discussed in the literature so far. Practitioners who have been experimenting with similarities across words and documents find that–at least on a Wikipedia corpus–the closest results for words are mostly other words, and for documents mostly other documents[3]. Furthermore, it has been stated that it depends on the training method and data whether it is meaningful to compare word vectors and document vectors[3]. In this paper, we assume that comparability is given, although word vectors and document vectors might be of slightly different nature.

4. Evaluation

We evaluate our approach with a large text corpus of a nation-wide public radio broadcaster in Germany that covers 63,165 manuscripts with about 70 million words in total. Each document has a minimum length of 100 words and is written in German language. 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 keywords are embedded in a company specific taxonomy that already lasts for decades and slowly changes over time. The taxonomy consists of 12,236 keywords on seven levels. On average, an archivist assigns 4.7 keywords to a document, in most cases between 3 and 7.

As text preprocessing, we replaced all capital letters to small letters, all numbers by their word equivalents (“1” to “one”), replaced all special characters and eliminated all punctuation, as is common for word2vec preprocessing. Also, we eliminated stop words and identified bigrams in the text corpus and in the taxonomy and concatenated the bigram words with underscores accordingly.

For each document vector, we calculated the dot products with all word vectors of the taxonomy, resulting in 63,165 x 12,236 dot products. The best dot product for each document represents the best prediction for the

[3] https://groups.google.com/forum/#!topic/gensim/Fujja7aOH6E

document. For each document, the calculated dot products show that the similarities drop drastically within the best 10 dot products. Figure 3 shows the sequence of best dot products for two example documents in its entirety (top) and for the top 100 dot products (bottom). Figure 4 shows the distribution of the top1 (best dot product per document) for all 63,165 documents.

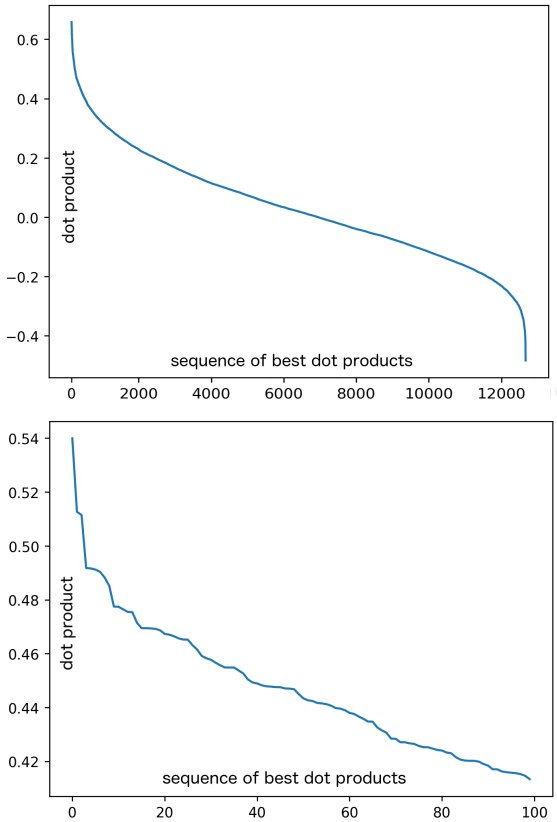

Figure 3. Sequence of best dot products for two example documents in its entirety (top) and for top 100 dot products (bottom)

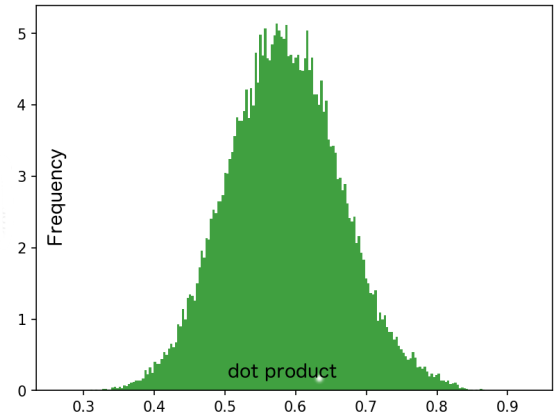

Figure 4. Distribution of best dot products

High dot products might indicate semantic similarity between documents and taxonomy word vectors; however, they are no guarantee. In contrast, low dot products are certainly no good ground to find keywords that describe the documents. Figure 3 indicates that the approach allows us to predict a handful of keywords with high dot products in the first drop-off of the graphs.

4.1. Evaluation Method and Metrics

For evaluation, we use the most common evaluation metrics for multi-label classifiers–recall and precision. We evaluate our approach in two perspectives, a) under a "perfect rationality" assumption and b) under a "bounded rationality" assumption, terms coined in a decision-making context by Simon [20].

Evaluation under "Perfect Rationality" Assumption. In decision tasks, human decisions are often considered as the reference for a machine's output. The basic assumption is that archivists decide under perfect rationality which keywords fit to the text and which not. In this view, machines can get close to the quality of human work but cannot beat it. The view of "humans do best" is often pursued in automation tasks that aim to imitate or replace human work without necessarily improving it. In practice, a rich data set of historical data is often available that can be used to evaluate machine against human. In our case, we have manually assigned keywords for 63,165 documents at hand that we can use for evaluation.

Evaluation under "Bounded Rationality" Assumption. In reality, archivists might not always be perfectly rational in their decisions. Time pressure, extensive knowledge of the taxonomy with 12,236 keywords, experience, fatigue and daily condition may influence the information processing of archivists and, as a result, the quality of their work. Other than in an evaluation under "perfect rationality" assumption, we assume that humans may miss out keywords or fail to assign the best keywords due to their "bounded rationality" [20]. This opens up potential for algorithms to outperform humans or to complement humans in their work.

As our evaluation method, we chose to reassess the keywords for 100 random documents out of 63,165. In order not to be primed and unbiased towards the predicted keywords, we mixed the assigned keywords of the archivist together with the top 10 predicted keywords and scrambled their order, so we were not able to tell anymore which key words belong to whom. Then three researchers had to decide independently for all 100 documents whether the keywords in the set of mixed keyword matches the particular document (y) or not (n). After the assessment, we used majority vote to solve differences: If at least two researchers have opted for yes, the keyword was considered as a correct keyword, otherwise not. Subsequently, we were able to compute recall and precision for the algorithm.

Figure 5 depicts true positive, false positive, false negative, and true negative sets for both assumptions. The set of all correct tags changes depends on the assumption (shaded in grey in Figure 5). Whereas under "perfect rationality" assumption, archivists per assumption were unbeatable, in a "bounded rationality" scenario, both archivists and algorithms can be right or wrong. For "perfect rationality" assumption (Figure 5 left), the archivist has both a recall and precision of 100%, and the algorithms has to predict a subset of the archivists keywords. For "bounded rationality" assumption (Figure 5 right), additional keywords can be correct and allow the algorithm to complement or outperform human.

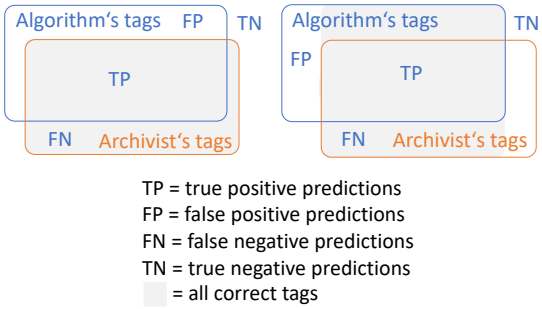

Figure 5. Sets for recall and precision under "perfect rationality" (left) and "bounded rationality" (right)

4.2. Evaluation Results

We evaluate precision and recall depending on the number of predicted tags. For "perfect rationality", Figure 6 (left) shows precision and recall as a function of the number of predicted tags. We see that the sweet spot for both recall and precision lies near 5 predicted tags and can be explained as follows: If we predict more words than the archivist, the superfluous words will be incorrect by assumption ("perfect rationality") and the precision decreases.

In contrast, if we predict less words than the archivist, we have insufficient recall by assumption. The point where recall and precision intersect and have their tangent point is almost exactly 4.7, the number of keywords that archivists assign on average.

Assuming "bounded rationality", we achieved much better precision results. Interestingly, the precision stayed on a more or less constant level of about 23% independent of the number of predicted tags. The recall however increased constantly with the number of predicted tags up to a level of 23% for 10 predicted tags.

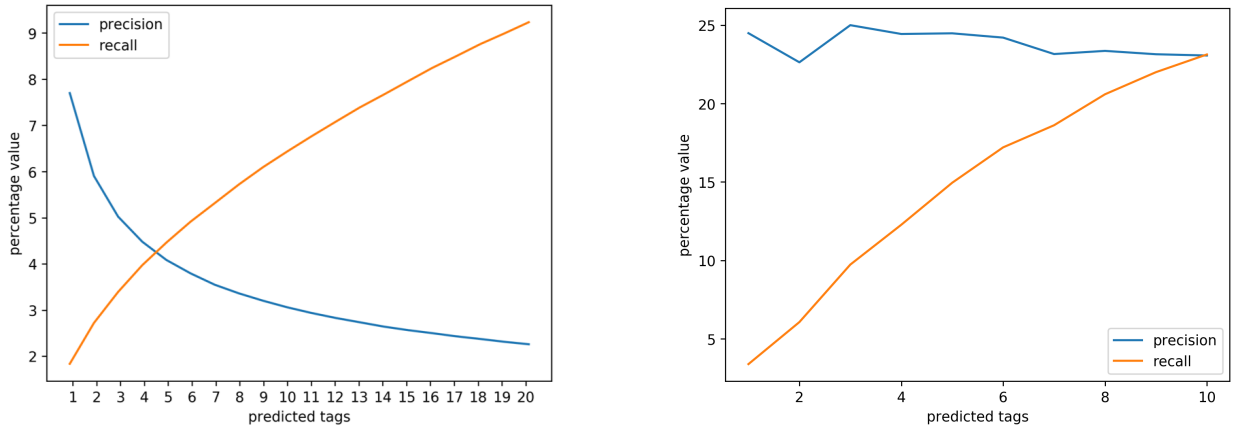

Figure 6. Precision and recall depending on number of predicted tags with (top) "perfect rationality" assumption (63165 documents) and (bottom) "bounded rationality" assumption (100 documents)

In Table 1, we depict three random examples of manually assigned keywords from archivists and the matching keywords identified by the algorithm.

Taking a second look at the generated keywords for the 100 documents, we find that the approach identified keywords that are not mentioned in the texts. Here we see the strength of this approach compared to syntactical approaches like tf-idf, because the context of words is more robust than syntax similarities.

However, the focus on context also brings along some drawbacks. During evaluation of the 100 documents, we noticed some keywords that did not match the content of the document but share a similar context with correct predictions. E.g., protestant church was predicted though the document was about catholic church only. Also, the algorithms could not differentiate very well between different music genres and predicted hip hop, rap, reggae, and punk together with traditional music and world music.

Table 1. Example of manually identified keywords and matching keywords identified by the algorithm (keywords translated into English)

Manually assigned keywords from archivists	Matching keywords identified by the algorithm
power economy, nuclear energy, energy and water management, nuclear phaseout, energy policy	power economy, nuclear energy, nuclear phaseout
church, protestant church, catholic church, work, religion, pilgrim	protestant church
music, new music, musical theater, opera, debut performance, composer	musical theater, opera

Obviously, the approach loses its discriminatory power in deeper levels of the taxonomy. The more specific the keywords in the taxonomy, the less accurate the prediction. As the word2vec approach represents word by their contexts, it is likely that words with similar contexts are predicted as well, even though they are incorrect. This seems to be an inherent characteristic of the approach but may be partly overcome with a larger document corpus that would enable a more differentiated training of contexts. Also, increasing the vector size that reflects the dimensionality of the word context may add discriminatory power.

4.2. Relaxing the Evaluation Allowing Parent Keywords

So far, we evaluated exact matches. However, in some cases, it might suffice to correctly predict a more general term. It is imaginable that an archivist tagged "catholic church", while the algorithm suggested "church" as a keyword, or the other way around. Therefore, in a relaxed evaluation, we also allow keywords on the parent level in the taxonomy both for predicted and archivists' keywords.

As a result, we obtain almost twice (2x) as many predictions (top 10 predicted words + max. 10 of their parental words) and twice as many correct keywords to be matched (archivist's keywords + their parental words). This way, under the "perfect rationality" assumption, we managed to obtain 248,049 correct predictions, leading to a precision of 248,049 / (63,165*5.2*2) = 38% and a recall of 248,049 / (63,165*10*2) = 19.6%.

By allowing matches in the next hierarchy level, we were able to increase precision and recall drastically. In fact, we could correctly predict 248,049 in 63,165 documents, which equals 3.9 correct keywords per document on average.

5. Discussion, Limitations and Further Research

We proposed an approach to identify taxonomy keywords for document corpora with the help of word embeddings. In the same way as Le and Mikolov stated, that "an important advantage of paragraph vectors is that they are learned from unlabeled data and thus can work well for tasks that do not have enough labeled data" [12], we pursued this advantage and extended it to unsupervised keyword identification for existing documents. Our results indicate that the approach has several advantages over supervised classifiers: no labelled training data is needed for supervised classification, and computational complexity is low (training of a shallow neural network and calculating dot products). Also, our approach has advantages over other unsupervised methods like tf-idf and TextRank, as we can predict words that do not appear in the respective document but directly match the vocabulary of a company's taxonomy.

Comparing recall and precision values on a just quantitative scale, at first glance, the achieved performance values do not seem to be competitive to what supervised classifiers usually achieve, given either a huge dataset or a by magnitude smaller number of classes. For 12,236 classes and 63,165 training documents however, also supervised approaches will not achieve incredible values but still have the drawback that they still need document-tag relations as training data. The comparison to other unsupervised classification approaches shows that our approach is in similar magnitude (around 30%) to what has been achieved on other datasets [4] (therefore having limited comparability).

Our evaluation showed that the approach was able to generate correct keywords that archivists did not assign. The approach may therefore complement the work of human annotators by giving suggestions to an expert which taxonomy keywords to consider. The suggestions can augment the work of the expert and speed-up his decisions on keywords that directly match the company-specific taxonomy. In a human-machine collaboration, both human and machine together can achieve better keyword quality and quality.

Considering further research, we still need to tweak our approach, for which we see two ways, a) optimizing the hyperparameters for training, and b) also considering parent keywords of the predicted keywords, making use of the taxonomy's hierarchy. This way, we want to address the issue of lacking differentiation in deeper taxonomy levels and to increase precision. After optimizing our approach, we still need to evaluate it against other unsupervised methods and on other data sets.

References

[1] R. Y. Wang and D. M. Strong, "Beyond Accuracy: What Data Quality Means to data Consumers," Journal of Management Information Systems, vol. 12, no. 4, pp. 5–33, 1996.

[2] A. Hulth, “Improved Automatic Keyword Extraction Given More Linguistic Knowledge,” in Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Stroudsburg, PA, USA, 2003, pp. 216–223.

[3] R. Baeza-Yates and B. Ribeiro-Neto, Modern Information Retrieval, 2ed edition. New York: Addison Wesley, 2010.

[4] R. Mihalcea and P. Tarau, “TextRank: Bringing Order into Text,” presented at the Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing, 2004, pp. 404–411.

[5] X. Wan, J. Yang, and J. Xiao, “Towards an Iterative Reinforcement Approach for Simultaneous Document Summarization and Keyword Extraction,” in 45th Annual Meeting of the Association of Computational Linguistics, 2007.

[6] W. Wu, B. Zhang, and M. Ostendorf, “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, Stroudsburg, PA, USA, 2010, pp. 689–692.

[7] X. Zuo, S. Zhang, and J. Xia, “The enhancement of TextRank algorithm by using word2vec and its application on topic extraction,” J. Phys.: Conf. Ser., vol. 887, p. 012028, Aug. 2017.

[8] T. Hofmann, “Probabilistic Latent Semantic Analysis,” in Proceedings of the Fifteenth Conference on Uncertainty in Artificial Intelligence, San Francisco, CA, USA, 1999, pp. 289–296.

[9] D. M. Blei, A. Y. Ng, and M. I. Jordan, “Latent Dirichlet Allocation,” The Journal of Machine Learning Research, vol. 3, pp. 993–1022, Mar. 2003.

[10] S. Debortoli, O. Müller, I. Junglas, and J. vom Brocke, “Text Mining For Information Systems Researchers: An Annotated Topic Modeling Tutorial,” Communications of the Association for Information Systems, vol. 39, no. 1, Jul. 2016.

[11] T. Mikolov, K. Chen, G. Corrado, and J. Dean, “Efficient Estimation of Word Representations in Vector Space,” in ICLR Workshop, 2013.

[12] Q. Le and T. Mikolov, “Distributed Representations of Sentences and Documents,” in International Conference on Machine Learning, 2014, pp. 1188–1196.

[13] J. Mitchell and M. Lapata, “Composition in Distributional Models of Semantics,” Cognitive Science, vol. 34, no. 8, pp. 1388–1429, 2010.

[14] F. M. Zanzotto, I. Korkontzelos, F. Fallucchi, and S. Manandhar, “Estimating Linear Models for Compositional Distributional Semantics,” in Proceedings of the 23rd International Conference on Computational Linguistics, Stroudsburg, PA, USA, 2010, pp. 1263–1271.

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

[16] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, “Distributed Representations of Words and Phrases and their Compositionality,” in Advances in Neural Information Processing Systems 26, C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, Eds. Curran Associates, Inc., 2013, pp. 3111–3119.

[17] R. Socher, C. C.-Y. Lin, A. Y. Ng, and C. D. Manning, “Parsing Natural Scenes and Natural Language with Recursive Neural Networks,” in International Conference on Machine Learning, 2011, pp. 129–136.

[18] J. Lilleberg, Y. Zhu, and Y. Zhang, “Support vector machines and Word2vec for text classification with semantic features,” in 2015 IEEE 14th International Conference on Cognitive Informatics Cognitive Computing (ICCI*CC), 2015, pp. 136–140.

[19] J. H. Lau and T. Baldwin, “An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation,” presented at the Proceedings of the 1st Workshop on Representation Learning for NLP, 2016, pp. 78–86.

[20] H. A. Simon, “Theories of Decision-Making in Economics and Behavioral Science,” The American Economic Review, vol. 49, no. 3, pp. 253–283, 1959.