Topic Embeddings – A New Approach to Classify Very Short Documents Based on Predefined Topics

Lasse Lommel\(^1\), Meike Riebeling\(^1\), Burkhardt Funk\(^1\) and Christian Junginger\(^2\)

\(^1\) Leuphana University Lüneburg, Institute of Information Systems, Lüneburg, Germany
{lasse.lommel,meike.b.riebeling}@stud.leuphana.de, burkhardt.funk@leuphana.de
\(^2\) Otto GmbH & Co KG, Data Science, Hamburg, Germany
christian.junginger@otto.de

Abstract. Traditional unsupervised topic modeling approaches like Latent Dirichlet Allocation (LDA) lack the ability to classify documents into a predefined set of topics. On the other hand, supervised methods require significant amounts of labeled data to perform well on such tasks. We develop a new unsupervised method based on word embeddings to classify documents into predefined topics. We evaluate the predictive performance of this novel approach and compare it to seeded LDA. We use a real-world dataset from online advertising, which is comprised of markedly short documents. Our results indicate the two methods may complement one another well, leading to remarkable sensitivity and precision scores of ensemble learners trained thereupon.

Keywords: topic modeling, word embeddings, LDA, seeded LDA

1 Introduction

With the increasing amount of textual data, the interest in text analysis has grown significantly in recent years and methods like topic modeling have become an integral part of research in computer science and information systems [1-3]. Topic modeling aims at finding topics in a collection of text documents.

One popular unsupervised method in the field is Latent Dirichlet Allocation (LDA), which is a probabilistic model that infers a document's topic based on the distribution of words therein, building upon the underlying assumption that topics generate words, which in turn generate documents. It was developed from probabilistic latent semantic analysis (PLSA) by Blei et al. [4] and has since been one of the most widely adopted approaches to topic modeling, which is mirrored by its readily-available implementations in many programming languages and environments.

However, in some scenarios, one may — possibly due to domain knowledge — already know about a specific set of topics that is present within the corpus or that one would like to classify documents into for other reasons. Under such circumstances, unsupervised methods may fail to identify particularly those envisioned topics in a corpus of documents. Standard LDA, for example, takes input on only the number of topics to
discover but it does not necessarily find topics similar to those that one aims to classify into. Instead, the topics identified with the means of standard unsupervised methods would have to be further investigated by a human in order to identify the discovered topics’ meaning, beyond them constituting a collection of associated documents, and possibly map them to a predefined topic scheme.

Supervised methods do not suffer from this problem since the labels establish the set of topics that can be discovered. Approaches of supervised learning suitable for topic modeling include the naive Bayes algorithm, support vector machines or decision tree models, which require attaching topic labels to words [5]. But also derivatives of LDA, namely supervised or discriminative LDA approaches (sLDA, DiscLDA) [6], [7], have been developed. However, all supervised models come at the cost of labeled data, which may be substantial, if not prohibitive in some cases. Moreover, in case the set of topics that one would like to categorize into changes, again significant amounts of newly generated labels or re-labeling of the data are required.

A solution to solve the dilemma of combining unsupervised methods with a predefined set of topics may lie in seeded LDA, a recent extension to the traditional form of LDA [8].

We develop an alternative method that employs recent developments in word vector representations. Our proposed method matches current undertakings in the field, as, for instance, Schwaiger et al. [9] and Murawski and Bick [10] have dealt with similar problems of classifying short documents into predefined sets. Schwaiger et al. [9] analyze social media posts and develop a classification tool, utilizing a dictionary-based approach combined with a multinomial Naive Bayes algorithm. In a similar endeavor, Murawski and Bick [10] employ LDA to sort job advertisements of data professionals into job sub-categories within specific job titles. However, their purely unsupervised approach yielded results that had to then be manually linked to the given category framework [10]. Further examples of topic modeling implementations are presented by Müller and Brocke [11] or Geva et al. [12]. In both papers, the authors use LDA to retrieve topics from a collection of documents. Müller and Brocke [11] apply LDA to further develop categories for an app store by analyzing the short descriptions provided for each application. Following a similar procedure, Geva et al. [12] analyze tweets to find patterns in the posting behavior of users.

Extended approaches to retrieve topic models are presented by few authors. Xu et al. [13] propose a combination of LDA and Clustering methods as a novel method to incorporate text associations. The approach is twofold. To find topic associations, a clustering algorithm based on the pairwise proximity of topics is applied. These associations are then linearly combined with LDA to enrich topic retrieval [13]. A very different direction for enriching topic modeling research is presented by Eickhoff and Wiencke [14]. Emphasizing the need to make sense of topics retrieved by topic modeling algorithms, they propose a mixed methods approach combining qualitative coding and quantitative clustering to decontextualize and evaluate topic models [14]. These examples underline the current focus on vanilla LDA, indicating a need for the development of specialized methods in topic modeling [1], [2].

We contribute to topic modeling research a novel alternative to (seeded) LDA, departing from the proven paths of generative probabilistic models. Our proposed method,
which we present in section 2, is easy to set up and allows high performance classification of (very) short documents into predefined topic schemes and does not require labeled training data. In section 3, we verify our proposed approach on a dataset of 380,000 URL-based short documents that we obtain from an online retailer, where the classification of such documents into a predefined taxonomy allows for a better understanding of the websites on which online advertisement is served. Section 4 discusses the results, which suggest the proposed method may constitute a valuable addition to the existing topic modeling tool set. Section 5 concludes.

2 Methods

Section 2.1 provides an overview of related work from the field of (seeded) LDA. Subsequently, we present our novel approach to topic modeling which relies on word vector embeddings (section 2.2). We refer to this approach as topic embeddings.

2.1 Seeded LDA

Seeded Latent Dirichlet Allocation is a powerful probabilistic model in the field of unsupervised topic and text mining algorithms. It is an extension of the standard LDA model developed by Blei, Ng and Jordan [4]. In contrast to vector and matrix based text mining approaches such as latent semantic indexing or clustering, LDA algorithms not only reduce dimensionality, but also account for documents being composed by a mixture of topics [4], [15], [16]. The method is therefore suited for organizing large and unstructured text documents by extracting representative topic compositions [17]. Successful implementations are documented in a broad range of domains. In the field of information system research, examples include the analysis of social media to develop a tool categorizing social media activities or the analysis of computer science job advertisements, which we described in section 1 [9], [10], [15].

Jagarlamudi, Daumé and Udupa [8] developed the seeded LDA model for settings where true labels are not available but prior knowledge of topics inside the corpus can be used to steer the resulting topic assignments towards predefined topics. The general idea of seeded LDA is to feed the model with information about the nature of the topics to look for by integrating seed words. Setting such seeds in the LDA algorithm aims at guiding the extraction of topics by skewing the distributions on the document level as well as on the word level (document-topic distribution, topic-word distribution).

On the topic-word level, seed words enhance the probability of a topic to generate words which are associated with the seed words. The model is then extended by another topic-word distribution for each seed topic. Hence, a new layer is added, where each topic is assumed to be a mixture of its regular topic distribution and the related seed topic distribution. In addition to the topic-word level, seeded LDA guides the probability distributions in the document-topic layer. For each set of seeds, a distribution over the regular topics is built, which acts as a prior for drawing the document-topic distribution. Drawing a topic for each document is enhanced by an extra layer, where the
first step is sampling a set of seeds and then using the corresponding distribution over topics as prior information to select the document topic mixtures [8].

In other words, seeded LDA is an unsupervised approach to direct topic mining towards a given set of categories. That is to say, it is a method which incorporates knowledge about desired labels in an unsupervised learning algorithm [8], [18]. However, it relies on a sufficiently large number of words per document to properly find topic models within a document collection. The algorithm is based on the probability of words to co-occur. As a result, the success of seeded LDA depends on the composition of the document collection and the quality of seed words, which not only have to differentiate a given topic from other topics but also must frequently occur in the corpus. However, for short documents or underrepresented topics, seeded LDA has its limitations.

2.2 Topic Embeddings

Word embeddings are high-dimensional representations of words in vector space. The words are translated into vectors of real numbers in such a way that semantically similar words result in similar vectors, where the translation is based upon evaluating how words co-occur with other words. Even though the basic idea of representing semantic similarities of words in vectors dates back to the 1960s, when the first attempts were made to use feature representations to capture such similarities between words [19], the interest in word embeddings was boosted recently when Mikolov et al. [20] introduced a neural networks-based model called word2vec in 2013, which permitted much faster training than previous models.

As words are translated into vectors, it becomes possible to determine numeric distances between words. Thus, one can calculate the distances between the words of a document and another term. This other term (or a collection of terms for that matter) could well constitute a topic, which leads to the core of our approach. It assigns each document to the closest topic in a given set of topics, where closeness is measured in the word embeddings’ vector space. Thus, we dub our novel classification method topic embeddings.

However, not all words in a document are of equal importance. Stop words like “the” and other recurrent words add no value to determine a document’s topic as they are neither unique nor characteristic to documents of any specific topic. To filter for the characteristic words that help to measure the distance between documents and topics, we use the term frequency inverse document frequency (tf-idf) metric. Tf-idf assigns weights to words within a corpus’ documents and balances the relative frequency of a word within a document against the word’s relative frequency in the whole corpus. High scores stand for characteristic words of a document which occur frequently (that is at least once) in the given document but relatively seldom in the corpus. To efficiently use tf-idf scores in our setting, we first compute a (sparse) document-term matrix that contains the tf-idf scores and normalize it row-wise so that the tf-idf scores per document sum up to 1, which makes them a suitable mean of weighting.

To assign a given document to a topic from a predefined set, we combine word embedding distances with tf-idf scores as follows: We measure the distances between all
words in a given document and the terms describing a topic and combine these into a weighted sum, where the weights are contributed by the document words’ normalized tf-idf scores.\(^1\) We repeat this for all topics and assign the document to the closest topic:

\[ \text{Define } \text{ClosestTopic}(x, T, \mathcal{C}, F) : \]

\[ \text{Require: } x, \text{ the input document with documents words } x^{(i)} \]

\[ \text{Require: } T, \text{ the set of topics, consisting of topics } t^{(j)} \text{ and topic words } t^{(j,k)} \]

\[ \text{Require: } \mathcal{C}, \text{ the corpus, comprised of a set of all documents} \]

\[ \text{Require: } F(\text{word, word}), \text{ a distance function from a word embedding space} \]

Calculate tf-idf document-word matrix \( W \) from \( \mathcal{C} \).

Normalize \( W \) row-wise (per document) and extract \( w \) with weights \( w^{(i)} \) for each document word \( x^{(i)} \).

\[ \text{for } j \text{ from } 1 \text{ to } |T| \text{ do} \]

\[ \text{distance}_j = \sum_{\text{len}(x^{(i)})} \sum_{\text{len}(t^{(j,k)})} F(x^{(i)}, t^{(j,k)}) \]

\[ y \leftarrow \text{argmin}(\text{distance}_1, \text{distance}_2, ..., \text{distance}_{|T|}) \]

\[ \text{Return topic } t^{(y)} \]

Words that are not in the vocabulary of the employed word embeddings model cannot be used, even though they may have high tf-idf scores. Documents that do not contain a single word from the word embedding model’s vocabulary can thus not be classified at all.

Our approach shares elements with the work of De Boom et al. [21], who devise vector representations of fixed-length short documents with the help of idf-scores and word embeddings. Other approaches to obtain text representations include deep learning and skip-gram based methods. Multilayer perceptron and convolutional neural networks can be used to arrive at sentence representations as De Boom et al. [21] note. However, as De Boom et al. [21] point out, these methods necessitate inputs of the same length or aggregation operations. Other techniques to measure semantic sentence relationships by vector representations include skip-gram inspired Paragraph2vec (Le and Mikolov [22]) and RNN-backed methods introduced by Kiros et al. [23]. However, documents may oftentimes rather constitute a collection of words than collections of actual sentences as our use case shows. Moreover, most of these techniques require retraining when changing the word embeddings or when encountering unseen examples as De Boom et al. [21] point out.

Our method, however, can employ pre-trained word embedding models without retraining while also being able to handle documents with unknown words to a certain extent, as the following sections will show. Also note that our method does not require any labeled training pairs of documents and topics.

\(^1\) Note that tf-idf-weighting topic descriptions may be useful if topic descriptions have words in common.
3 Use Case and Benchmark

The evaluation of advertisement effectiveness is a common undertaking in managing online retailers. For example, retailers may wish to analyze the content of ad serving websites that a web shop’s audience is frequenting. Categorizing these websites may be helpful to oversee ad spending and to gain insights about the type of audience that the online retailer is attracting. However, sorting into arbitrary categories is usually not helpful as it hinders comparisons with established taxonomies. Hence, classifying into predefined categories is a necessity, rendering the classic LDA approach unfeasible. Labels on the other hand would be costly to acquire in such a setting since they require human input.

We evaluate the predictive performance of our topic embeddings approach against the seeded LDA approach on a dataset of URLs obtained from a German online retailer. The dataset contains approximately 380,000 records of URLs with a .de top-level domain, from which the vast majority are German-content websites, as a qualitative scan of the data reveals. For the word embeddings model, we follow Müller’s procedure [24] to train a 300-dimensional German word2vec model. As data inputs, we use a corpora of German news articles from 2007-2013 [25] and a copy of the German Wikipedia (retrieved Dec 21, 2017).

In order to allow for a quantitative performance evaluation of both methods, we manually label a fraction of 2,488 randomly selected records, which we then randomly split into a test and holdout set in an 80:20 ratio. To get consistent labels, two authors labeled the first 500 documents together and consult each other in ambiguous cases during labelling of the remaining records.

3.1 Topic Taxonomy

We use the Google AdWords taxonomy to predefine 22 main topics [26], which are further divided into sub-, sub-sub and sub-sub-sub-categories. However, we restrict ourselves to classifying into the 22 main topics.

<table>
<thead>
<tr>
<th>Topic</th>
<th>Seed Words</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dining &amp; Nightlife</td>
<td>dining, nightlife, night club, disco, restaurant, menu, bar, drinks, DJ, party, dancing, going out</td>
</tr>
<tr>
<td>Computers &amp; Consumer Electronics</td>
<td>computer, consumer electronics, pc, laptop, mouse, keyboard, game console, playstation, xbox, wii, nintendo</td>
</tr>
<tr>
<td>Food &amp; Groceries</td>
<td>food, groceries, juices, soft drinks, alcohol, beverages, vegetables, fruits, pasta, rice, potatoes, supermarket, milk, cheese, eggs, butter, bread, household goods, detergents, consumer goods</td>
</tr>
</tbody>
</table>

Table 1. Examples of the used topics and their corresponding seed words.

2 The originally used seed words are German and have been translated.
As the taxonomy is in English and our documents are in German, we first translate the 22 main categories and their 252 sub-categories. To create further seed words, we additionally enrich the topic descriptions with extensions as well as with some associated entities and proper names. As extensions we consider synonyms as well as words (mostly nouns) associated with a given topic, despite them not being direct translations of the original English topic. As associated entities we consider popular brands or institutions linked to certain topics. Table 1 lists seed words for three example categories. Also, we make sure to use each word in the description of only one topic and avoid words which could be associated with more than one topic altogether. In total, the final topic taxonomy consists of 421 words in the translation part, 656 words as extensions and 49 words in the associated entities section.3

3.2 Preprocessing: Constructing Documents from URLs

We build our documents from the word tokens inside each URL’s host and path. However, in some cases, the URL is extremely short and does not provide much useful information. For example, the host may be a brand name or another non-word, combined with a very short or non-word only path. Thus, we use a web crawler to extend our URL documents by the corresponding website titles as well as description and keyword tags. We thus end up with a corpus of 384,285 documents with an average (and median) length of 24 words when building the documents from URL tokens as well as all three crawled information sources.

3.3 Grid Search: Optimization of the Hyperparameter Configuration

Our approach to classifying the URL documents is twofold and comprises of two stages: After the pre-processing stage in which the documents are build, we classify them with either one of the two methods described in section 2. Each stage includes several hyperparameters to be tuned, which influence the performance of the resulting models. To examine the influence of these parameters and to identify the best configurations for our two methods, we conduct a grid search. The specific configuration options are shown in Table 2.

<table>
<thead>
<tr>
<th>Area</th>
<th>Options</th>
<th>Range of values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Topics</td>
<td>Topic description sources</td>
<td>[Translations, Extensions, Entities]</td>
</tr>
<tr>
<td></td>
<td>Include sub-category words</td>
<td>[True, False]</td>
</tr>
<tr>
<td>Documents</td>
<td>Source of document contents</td>
<td>[URL Tokens, Crawling Tokens]</td>
</tr>
<tr>
<td></td>
<td>Crawling sources</td>
<td>[Title, Description, Keywords]</td>
</tr>
<tr>
<td>Seeded LDA</td>
<td>Seed Confidence</td>
<td>[0.2, 0.8]</td>
</tr>
<tr>
<td></td>
<td>Replace non-vocabulary seeds</td>
<td>[True, False]</td>
</tr>
<tr>
<td>Topic Embeddings</td>
<td>Weight words with tf-idf</td>
<td>[True, False]</td>
</tr>
</tbody>
</table>

3 Upon request we are happy to share the list with interested readers.
During pre-processing, the documents can be constructed from either only the words in the URL, the words obtained by the web crawler or combinations thereof. Specifically, we distinguish one URL word source, three crawling word sources (website title, description and keywords) and combinations thereof. This procedure also allows us to more closely examine the effects of including the additional information retrieved from the web crawler on our two models’ predictive performances. The second part of the pre-processing stage is the aggregation of information about the topics from our taxonomy. As described in section 3.1, the taxonomy was translated and then extended by characteristic words and associated entities. For both models we test the effect of including each of these sources as well as their combinations to guide the learning process. Further, we vary the amount of information included in the topic descriptions by either combining all information from sub-categories into their respective main categories, which we want to predict on, or rather leaving out the information related to the sub-categories altogether. Also, both our classification techniques require a number of hyperparameters to be set. The seeded LDA algorithm provides a variety of choices for the seed sets: We test two options for the degree to which the initial distributions are skewed towards the seed words, comparing a very high confidence of 0.8 against a low one of 0.2. Additionally, we test for the effects of replacing seed words that are not contained in the document corpus’ vocabulary with the help of the word embeddings model, by means of replacing the missing words with the most semantically similar in the corpus vocabulary. For the implementation of the topic embedding approach, we include the option of weighting the influence of each word in the document by the tf-idf score. In total, the grid search includes 384 hyperparameter combinations that we evaluate on the labeled test set.

![Figure 1](image.png)

**Figure 1.** Confusion matrices of sensitivity-best models (left: Seeded LDA, right: Topic Embeddings). See Table 3 for topic names.

---

4 We ignore combinations of topic description sources without translations since extensions and associated entities were less densely available and empty for some categories. Early trials indicated that using only extensions and/or associated entities leads to expectably poor results. Similarly, we exclude combinations of crawling sources that lack page titles because many crawled pages do not feature descriptions and/or keywords.
4 Results

The baseline for evaluating the results of the model is the effectiveness of simply predicting the most common category. To estimate this baseline score, we calculate the distribution of topics in the section of the dataset that we manually labeled. The most frequent label is "Vehicles" with a share of 18.8%. For the top-2 and top-3 scores the baseline is 30.9% and 42.3% respectively.

4.1 Seeded LDA Method

The seeded LDA approach correctly classifies a maximum of 45% of all instances in the test set in terms of sensitivity, which measures the share of correctly discovered topics, and reaches 55% in terms of precision, which indicates how useful the predicted labels are, i.e. how large the share of relevant labels is among the retrieved labels\(^5\). However, it shows poor performance on certain topics like “Food & Groceries”. Not using the web-crawled data decreases the performance by 13 percentage points, meaning that the sensitivity climaxes at 36% when the method only works on the tokens from the URL itself.\(^6\) With regards to the possible combination of sources that describe the topics (translations, extensions, associated entities), it becomes clear that the extensions make a significant difference as they add 10 percentage points to the performance, while the inclusion of associated entities has no impact. Replacing seed words not found in the document corpus’ vocabulary makes no difference in our case, which is likely due to the fact that already a very large portion of the topic descriptions vocabulary is shared by the document corpus vocabulary. Further, the seeded LDA model’s top-2 and top-3 predictions cover the correct topic in 62% and 69% of all cases respectively.

4.2 Topic Embeddings Method

The topic embeddings approach reaches scores of 41% in sensitivity and 49% in precision. While it performs very well on certain topics like “Vehicles” (21), it does not do well on the “Hobbies & Leisure” (10) category and, as obvious from the confusion matrix in Figure 1b, frequently confuses "Home & Garden" (11) with "Real Estate" (17). Interestingly, the best topic embedding model relies only on data from the main categories as topic descriptions and excludes extensions and associated entities.\(^7\) Adding the latter two or including the descriptions of sub-categories makes no difference aside from a stark increase in computation time as the number of required word to word distance calculations scales linearly with the length of the topic descriptions. The best

---

\(^5\) The optimal configuration includes most document word sources (URL tokens as well as title and keywords from crawling results) and requires all seed word sources from main and sub-categories as topic descriptions.

\(^6\) Topic descriptions again require all seed word sources and all words from main and sub-categories.

\(^7\) The optimal model configuration further includes building documents from URL tokens as well as website keyword and title tags.
model uses tf-idf to weigh words but the influence of using tf-idf is, perhaps due to the shortness of the documents, very low. Restricting the compilation of documents from using the data obtained from the web crawler results in only a small performance drop of 3 percentage points in sensitivity and 1 percentage point in terms of precision. The topic embeddings' top-2 and top-3 scores cover the correct topic in 50% and 56% of all cases respectively, both including crawled data for document generation.

Table 3. Performance summary of the topic embedding and seeded LDA models evaluated on the test set.

<table>
<thead>
<tr>
<th>Topic Embeddings</th>
<th>Seeded LDA</th>
<th>ΔSensitivity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Support</td>
<td>Sensitivity</td>
<td>Precision</td>
</tr>
<tr>
<td>0</td>
<td>Apparel</td>
<td>43</td>
</tr>
<tr>
<td>1</td>
<td>Arts &amp; Entertainment</td>
<td>119</td>
</tr>
<tr>
<td>2</td>
<td>Beauty &amp; Personal Care</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>Business &amp; Industrial</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>Computers &amp; Consumer Electronics</td>
<td>77</td>
</tr>
<tr>
<td>5</td>
<td>Dining &amp; Nightlife</td>
<td>18</td>
</tr>
<tr>
<td>6</td>
<td>Family &amp; Community</td>
<td>126</td>
</tr>
<tr>
<td>7</td>
<td>Finance</td>
<td>14</td>
</tr>
<tr>
<td>8</td>
<td>Food &amp; Groceries</td>
<td>103</td>
</tr>
<tr>
<td>9</td>
<td>Health</td>
<td>49</td>
</tr>
<tr>
<td>10</td>
<td>Hobbies &amp; Leisure</td>
<td>237</td>
</tr>
<tr>
<td>11</td>
<td>Home &amp; Garden</td>
<td>159</td>
</tr>
<tr>
<td>12</td>
<td>Internet &amp; Telecom</td>
<td>20</td>
</tr>
<tr>
<td>13</td>
<td>Jobs &amp; Education</td>
<td>59</td>
</tr>
<tr>
<td>14</td>
<td>Law &amp; Government</td>
<td>23</td>
</tr>
<tr>
<td>15</td>
<td>News, Media &amp; Publications</td>
<td>90</td>
</tr>
<tr>
<td>16</td>
<td>Occasions &amp; Gifts</td>
<td>9</td>
</tr>
<tr>
<td>17</td>
<td>Real Estate</td>
<td>121</td>
</tr>
<tr>
<td>18</td>
<td>Retailers &amp; General Merchandise</td>
<td>250</td>
</tr>
<tr>
<td>19</td>
<td>Sports &amp; Fitness</td>
<td>67</td>
</tr>
<tr>
<td>20</td>
<td>Travel &amp; Tourism</td>
<td>51</td>
</tr>
<tr>
<td>21</td>
<td>Vehicles</td>
<td>367</td>
</tr>
<tr>
<td>total / average</td>
<td>2036</td>
<td>0.41</td>
</tr>
</tbody>
</table>

4.3 Ensemble Learning: Combining Topic Embedding and seeded LDA

By comparing the two confusion matrices in Figure 1, one can easily discover that the two methods complement each other in parts. Thus, we train an ensemble learner whose results are depicted in Figure 2.
We combine the predictions of our two approaches in a decision tree, where the labels obtained during the grid search outlined in section 3.3 serve as features. We limit the resulting tree to take only two features as input due to the high computational cost of each feature, which we assume could otherwise be prohibitive in a realistic application scenario. We train a tree with all features and select the two most important ones. Not surprisingly, each of our two models contributes one feature. While the selected topic embeddings model is the best performing one described in section 4.2, the selected seeded LDA model is very similar to the best-performing one described in section 4.1 and only differs with respect to including the crawled website descriptions into the documents. Subsequently, we undertake an additional optimization endeavor with three-fold cross validation to identify the optimal tree in terms of test score, which is shaped by using entropy as a split criterion, an impurity decrease of 0 and an optimal depth of 13. We decide to further decrease the maximal depth to 9 to prevent overfitting and reduce complexity. We evaluate this final ensemble model on the holdout-set and reach scores of 65% sensitivity and 70% precision.

![Figure 2. Left: Confusion matrix of ensemble learner evaluated on holdout-set (452 documents, none in topic 16). Right: Results of the grid search for the final ensemble learner.](image)

Table 4. Performance summary for seeded LDA and topic embeddings evaluated on test-set and final ensemble learners evaluated on holdout-set.

<table>
<thead>
<tr>
<th>Options</th>
<th>Crawling Sensitivity</th>
<th>Crawling Precision</th>
<th>No Crawling Sensitivity</th>
<th>No Crawling Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Seeded LDA</td>
<td>45%</td>
<td>55%</td>
<td>36%</td>
<td>49%</td>
</tr>
<tr>
<td>Topic Embeddings</td>
<td>41%</td>
<td>49%</td>
<td>38%</td>
<td>48%</td>
</tr>
<tr>
<td>Ensemble</td>
<td>65%</td>
<td>70%</td>
<td>50%</td>
<td>56%</td>
</tr>
</tbody>
</table>

Note: Topic word sources vary by method.

8 See Table 4 for results of a crawl-data free ensemble learner fitted by the same procedure.
5 Discussion

We introduced a novel method for unsupervised topic modeling with predefined topics and evaluated its performance against the established seeded LDA model in a real-world application setting characterized by very short documents.

Our results emphasize that the seeded LDA method is very dependent on the length of its inputs and greatly benefits from additional data, both in terms of document length and in terms of topic description length. Since sufficiently long documents may not always be readily available as the case study demonstrates, this is one of the main shortcomings of the seeded LDA model. Nonetheless, it does perform very well when supplied with the required text volume. The topic embeddings model on the other hand does not profit from increased document length or topic descriptions. Its greatest strength is its robust performance that is almost on par with the seeded LDA model in a data-sparse environment of very short documents. The topic embeddings approach could easily be extended and possibly be refined in further research, for example by accounting for the varying performance of word embedding models depending on the word types. Rubinstein, Levi, Schwartz and Rappoport [27] show that on the one hand, word embedding models are limited with regards to learning attributive relationships like that between the words “fast” and “jet” while on the other hand they emphasize the strengths of such models regarding taxonomic relationships like that between “car” and “vehicle”. This hints at potential for performance improvements of the topic embeddings approach by only considering nouns while dropping everything else within a document. Another possible field of study lies in evaluating the effects of including bigram tokens in the model and, more far reaching, evaluating the robustness of the performance with regards to the chosen seed words. In addition, it would be interesting to compare the topic embeddings model performance against traditional classification methods in supervised settings. We suspect our method would naturally compete well against established models when training data is sparse since it has shown strong performance without learning from labeled examples, as outlined in section 4.2. However, with increasing amounts of training data, we believe it would be outperformed by traditional bag of words classifiers or other word embedding approaches like Facebook research’s fastText [28], which averages word embeddings of a document and passes the result through a softmax layer. While supervised classifiers obviously benefit from additional training data, our approach is independent of the amount of training data. It would be helpful to further investigate at which amount of data supervised learning algorithms take over.

However, one must keep in mind that both seeded LDA and the topic embeddings method are only unsupervised with regards to not requiring labeled document data. They obviously require input on the topics which one would like to classify into. Further research could possibly refine this area by investigating optimal discovery and selection strategies for seed words as well as determining which amount of information is optimal for maximum performance.

By combining both methods into an ensemble model, we showed how their individual strengths can complement each other. However, this requires one to pay the high
price of labeling some data that essentially renders the whole undertaking a partly supervised method. Moreover, even the ensemble model struggles with very broad categories like “Hobbies & Leisure”. Such diverse topics are problematic and further research is necessary to evaluate their handling, possibly by breaking them down into subcategories for classification and re-aggregating them afterwards.

Notable work on the combination of LDA and word vectors has been published in the past, e.g. by Nguyen et al. [29] and Moody [30]. While these combinations show strong performance, they lack our approach’s ability to be steered towards a given taxonomy. Nonetheless, it would be valuable to investigate whether these existing models can be amended to allow for guiding towards given sets of categories.

To conclude, the topic embeddings model that we developed proves to be a valuable addition to current topic modeling research. Its relative independence of document length distinguishes it markedly from LDA approaches. Together with the ability to classify into given categories and its unsupervised nature, it marks one such specialized topic modeling technique that Eickhoff and Neuss [1] called for. Its strong performance on very short URL documents hints at further use cases in other areas where short documents are present, such as social media data or item descriptions in online shops. However, as language differs by context and evolves over time, one should keep in mind that the backbone of the approach, the word embeddings model, may need to be retrained on corpora of current, application-relevant text.

References


