ISSN 2395-8618

Zero-Shot Learning for Topic Detection in News Articles

Björn Buchhold and Jörg Dallmeyer

***Abstract*—We present a method to detect topics in news articles. The topics of interest are each represented by a descriptive document. We train a model that can be seen as a similarity function between such a descriptive document and a news article. Our model is a neural network that operates on two kinds of inputs. (1) The full texts of the descriptive documents and the news articles are passed through the same recurrent encoder network and then the distance of the resulting encodings is taken. (2) Our proprietary NLP pipeline and knowledge base are used to recognize named entities and significant keywords and we compute features based on their overlap for a descriptive document and a news article. Our model finally combines the encoding distance with the overlap features and acts as a binary classifier. We evaluate and compare several model configurations on two datasets, a large one automatically created from Wikipedia and a smaller one created manually.**

***Index Terms*—Topic detection, zero-shot learning, NLP, deep learning.**

I. Introduction

THIS WORK focuses on a special kind of topic detection. Think of a topic that emerges in news, for example the *Diesel emissions scandal*. It can be of great value to recognize all news articles that talk about such a topic. In combination with the identification of named entities, users can then track if a company they are interested in (maybe because they own its stock; maybe because the company is one of their suppliers, customers or a competitor) is mentioned in the context of the scandal. Likewise, the other way round is useful, i.e., to see all topics that are emerging for a selection of interesting entities.

This task is challenging in several ways: topics are often not mentioned literally in the news but inferred by the content and the involved entities. In the example of the *Diesel emissions scandal* not all three words have to be mentioned in this order for a news article to be talking about the topic, however, simply looking for the individual words, variants and synonyms is hopelessly imprecise. Furthermore, classic approaches to topic modeling are not applicable here either, because they all build models around a corpus that already contains the topics of interest. However, it is the very essence of news, that entirely new topics emerge and are reported on.

We want to improve an existing system performs for news monitoring.

Manuscript received on June 18, 2019, accepted for publication on September 4, 2019, published on December 30, 2019.

The authors are with CID GmbH, Germany (e-mail: {b.buchhold, j.dallmeyer}@cid.com; web: http://cid.com).

It performs web crawling, content extraction, and several NLP tasks including the identification of named entities from customizable knowledge graphs. Enriched documents are stored in an indexing system with support for near real-time analyses on millions of documents. So far, the system supports two approaches for modeling and detecting topics, which tackle the problem from different perspectives.

So-called Hot Topics (HT) are computed on a document analysis set (e.g., over the last three days). Entities and keywords that are significantly more frequent in the analysis set than in a reference set are grouped according to co-occurrence. Such a group then represents an HT. An HT is dynamic, unsupervised and has no meta information and no proper name.

The second approach consists of so-called Supervised Topics (ST). STs are manually defined topics consisting of a set of weighted keywords and named entities. With STs it is possible to detect topics in future news (e.g., earth quakes or C-Suite changes). STs use a linear retrieval model of weighted aspects. However, the fine-tuning of selected aspects, their weights and acceptance thresholds requires domain knowledge.

The contribution of this paper is a novel approach for the kind of topic detection described in the beginning. We base it on descriptive documents for topics that can then be associated to news articles using a combination of traditional NLP and Deep Learning: We view our problem as instance of zero-shot learning. We train an encoder that infers abstract representations from text documents and our NLP pipeline extracts named entities and keywords from them. For a descriptive topic document and a news article these abstract representations and the overlap in extracted NLP features are used to calculate their similarity. New topics can be introduced to the system without re-training the underlying model. We show how training data for this problem can be gathered automatically, introduce suitable models and evaluate them in different experiments.

We make use of the work by the Wikipedia[1] community, whose manual curation yields up-to-date topic documents with very high quality. These documents have a unique identifier, a description and representations in several languages. This gives control for selection and customization of relevant topics for users with diverse interests.

[1]https://en.wikipedia.org/wiki/Portal:Current_events

ISSN 2395-8618

Section II reviews related work. In Section III the fundamental task of generating training data for the learning step is discussed. We give a model description in Section IV. Section V shows the experimental setup and results, as well as variants of the model. We conclude in Section VI and give ideas for future work.

II. Related Work

This work studies a new problem as motivated in Section I. Thus, we cannot compare our results to those for well-studied problems with prominent datasets. Nevertheless, there are several fields of work that are closely related. We distinguish three categories: (1) Literal approaches and the well-established task of Named Entity Identification where topics could be treated as just another kind of entity; (2) Topic Models, which are well-studied but not compatible with the way we want to define topics of interest; (3) Classification via zero-shot learning in other domains, i.e., machine learning approaches that assign classes, even if there never was any training data for a particular class, e.g., systems for face recognition.

A. Literal Topic Identification

We could treat our problem as just another kind of Named Entity Identification with topics as entities to identify. In our experiments, this approach did not yield good results. While this works out in some cases, the limitations quickly become apparent.

Literally matching *Brexit* in news articles and maybe adding other strong signal words, e.g., *Brexiteer* works pretty well. A counter example is the *China–United States trade war* and a news article about the implementation of certain tariffs. It is conceivable that there is no literal mention of the trade war, even though the article is relevant. When too liberal signal words are added, the precision drops significantly.

B. Topic Models

Topic Models are well-researched. In a sense, they induce a soft clustering (where a document can belong to 30% to topic A and to 70% to topic B) on a document collection. Thus, these topics are usually purely statistical and abstract, i.e., they do not carry a name nor necessarily correspond to an intuitive topic.

The most prominent approaches to topic modeling include Latent Dirichlet Allocation (LDA) [1] and Probabilistic Latent Semantic Analysis (pLSA) [2]. Usually, these approaches are given a number of topics to discover. They then assume that documents are produced by the following generative process: For each word, first one topic of the current document is chosen, and then a word is picked from all possible words with the conditional probability given the chosen topic. Given this generative model, the topics are assigned to a document according to a maximum likelihood estimate.

An important difference to our problem is that only the number of topics to infer is given and a document collection is fit accordingly. The resulting topics are not necessarily interpretable in an intuitive way. More importantly though, newly emerging topics do not work at all unless the whole model is fit to a document collection in which the topic plays a significant role.

Labeled LDA (LLDA) [3] is an extension of LDA where not only the number of topics to infer is given but also concrete topic labels. This overcomes the problem of nameless topics that do match expected kinds of topics. However, this extension is also not applicable to our scenario, because it still has to be fit on a corpus that already contains all topics of interest.

In [4] the static Dewey Decimal Classification (DDC) is applied for thematic classification of short text snippets, e.g., tweets. The work deals with the need for fast and accurate classification in scenarios of lexical sparseness. The best classification results were achieved by a combination of a Support Vector Machine (SVM) and a Neural Network (NN). The SVM used *tf-idf* term weights, n-grams and LDA topic features. The NN uses fastText [5] in combination with nodes with an activation function signaling membership to LDA based topics.

The training is done on DDC classified German documents. The derived model is used to explore the distribution of DDC topics in a pool of text documents. In contrast to that, we focus on changes in the collection of topics.

The Topic Detection and Tracking (TDT) study [6] features several related tasks. However, most of them are similar to work discussed previously: Entire text corpora are segmented into topical clusters, or well-known topics are tracked throughout news. One task for *On-Line New Event Detection and Tracking* [7] is more closely related to our work, the major difference being the absence of our descriptive documents. However, the approaches to online detection still treat the problem as a clustering problem, where unassigned documents form their own topical clusters. The study concludes that "Online detection cannot yet be performed reliably". In a sense, our problem is a slightly simplified variant of this hard task.

C. Zero-Shot Learning

The problem looks like yet another instance of text classification. However, since new topics can occur at any time, the problem becomes a lot more intricate. We cannot expect to re-train the model every time a new topic occurs, but even more importantly, we have no training data to do so. Thus, the problem can be seen as instance of zero-shot learning, i.e., the model has to predict topics for which it has not once seen explicit examples to learn from. This is sometimes also known as zero-data learning [8].

In [9], the authors present a framework for zero-shot learning. Just like in our scenario, not all target classes

ISSN 2395-8618

occur in training data. However, there is a difference. Their classification targets are from a semantic knowledge base, which describes animals by features such as *is it furry?* or *does it have a tail?*. The authors then design their classifier to predict a feature vector with scores for the features from the semantic knowledge base. The class with the closest feature vector is then predicted as the result. In their case study, they have trained models to produce such vectors from fMRI images, i.e., from people's neural activity.

In our case, we do not have such a semantic knowledge base with all, and especially upcoming, topics. Our problem is more closely related to face recognition and face verification, where new faces are added without manually deriving feature representations (and also without re-training the models). A popular system for face recognition is FaceNet [10], which has inspired much of our work. The authors train a neural-network encoder that produces a high-dimensional encoding vector for each known image and for the input image. A threshold for distances in euclidean space between these encodings can then be used to accept or reject matches. Our work follows the same approach where we replace the image encoder with a textual encoder. We augment this model by features derived from our NLP pipeline. However, it is possible to train a textual encoder, so that the L2-distance between encodings already identifies topics fairly well. In our experiments in Section V we quantify this and compare the approach to other variants and to our full model.

An important part of FaceNet is that the encoders are trained using a specialized triplet-based loss function that operates on triples of anchor image, positive example and negative example.

The loss function then requires the encoding of the anchor to be closer to the encoding of the positive example than to the encoding of the negative example by a given margin. We can also use this loss function to train our textual encoders, however with slightly worse results than by simply training them in a binary classifier.

III. Acquisition of Training Data

For most practical applications of machine learning, finding a suitable model is just one part of the problem. In absence of an established dataset, the acquisition of training data is often the biggest challenge to overcome. For our use case, we are not aware of any suitable dataset and manual creation would be extremely tedious and costly. Hence, we have designed a system to automatically retrieve such data from publicly available information, in particular from Wikipedia.

Our source are Wikipedia articles about the current events for a given date. These pages exist for every day since 1994, and since 2003 they adhere to a format that is very useful for our purpose. For important events, editors quickly generate dedicated Wikipedia articles and link to them whenever there are new developments. The event essentially becomes a topic that occurs in the news for a few days, weeks, or possibly many years (e.g., major political conflicts).

In Figure 1 we illustrate how we extract topics for a specific day: We ignore category headlines and work with the bullet points. If (and only if) a top-level bullet point does have subordinate bullet points, we take the linked Wikipedia articles of the top-level bullet as topic The contents of these linked Wikipedia articles can then be used as descriptive documents. Further, every external link under the bullet point refers to a relevant news article. We extract the body text behind those links and regard the associated topic as a positive training example.

These connections to external news articles turn out to be very reliable training data. However, many of them have to be skipped: The links may no longer be working, they may be unavailable for automatic retrieval, the news articles may be in some other language, etc. To gather additional positive topic-news pairs, we also visit the Wikipedia articles about the topics themselves and extract the References section (which contains links to documents outside Wikipedia). This gives us numerous positive pairs, albeit with some uncertainty (a reference may be evidence for a very specific statement in the article and is not necessarily relevant to the entire topic). Experiments have shown that their inclusion does significantly more good than harm.

We treat our problem as pairwise classification problem and thus need negative training examples, i.e., news articles and topics which should not be detected for them. One way is to sample these pairs at random. For each positive pair of topic and news article, we can produce negative pairs with the same topic and a random other article and negative pairs with the same article and a random topic. We choose to make the number of negative pairs configurable for two reasons: (1) Taking all negative pairs for thousands of topics and tens to hundreds of thousands of articles results in way too many negative pairs to handle efficiently. (2) Such negative pairs are not perfectly reliable. For example, a news article may be linked to the topic *Dismissal of James Comey* but it may also be relevant to topics like *Presidency of Donald Trump*. For these two reasons we limit (and thus essentially under-sample) the number of pairs of negatives. We end up with sufficient data and while we may still pick a false negative pair at random, the chances are very low. At the very least we will have significantly more correct positive pairs (and correct negative pairs) than problematic pairs.

While the above strategy already produces a functioning model, a problem remains. Negative pairs picked at random have a tendency to be too easy to distinguish from positives ones, because they may be from very different domains. The resulting models do not work well in practice, despite achieving very high accuracy (and also great performance w.r.t. other metrics like F-measure) on our test data. As an example, assume a news article about the topic *2019 elections in India*. It is easy to accept that topic and reject completely unrelated ones like *Gun laws in New Zealand*. However, it is much harder to handle topics like *2014 Indian general election* or *2019 Sri Lankan presidential election* in that case, because

ISSN 2395-8618

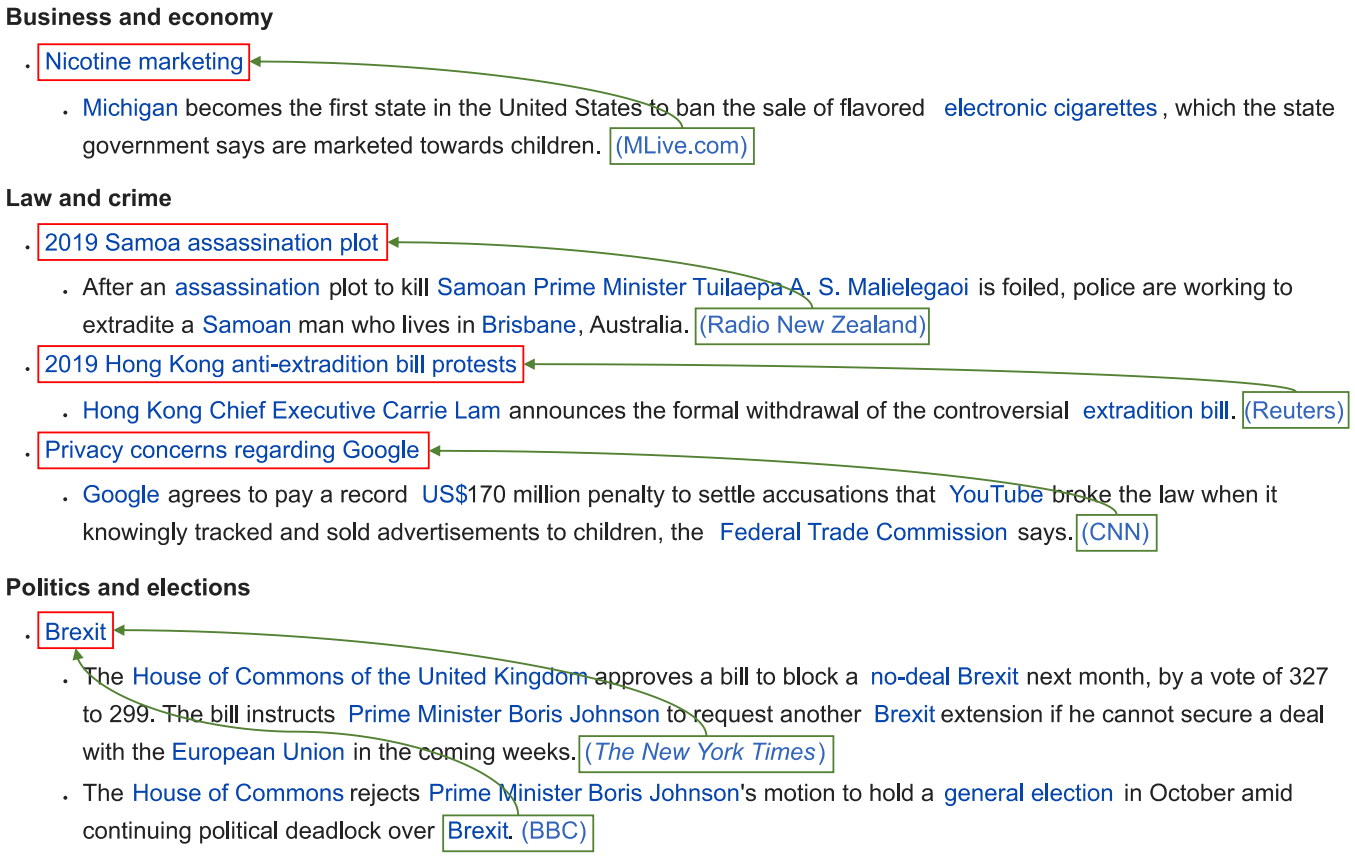

Fig. 1. An excerpt from the Wikipedia article on current events of September 9, 2019. The red boxes mark the topics we are able to extract from this. The green boxes mark corresponding relevant news articles.

these topics are semantically more closely related, yet still a wrong choice for the article. If we do not include these hard pairs, it becomes much easier to correctly classify the items of our evaluation set than to use the model in practice.

Therefore, we created a harder dataset by selecting random news articles and used early versions of our models to predict the 10 most likely topics for each. In essence, this gives us topics that are similar to the news article w.r.t. different notions of similarity but may or may not be correct. We then asked 17 judges to label the topic-article pairs. We kept labels on which enough judges agreed. In our experiments in Section V we elaborate further on this process and on our results.

IV. Model

We want to be able to recognize multiple topics for a single news article, hence our problem can be seen as a multi-label classification. Consequently, we design our model to take a pair consisting of a descriptive document for a topic and of a news article. Then the model acts as a binary classifier that decides to accept or reject the pair.

Our model is an ensemble Neural Network that consists of a binary classifier working with features based on entity- and keyword-overlap on the one hand, and a siamese text encoder NN (bidirectional LSTMs with self attention) on the other hand. We illustrate this in Figure 2. The encoder NN (shown as the top part) produces high-dimensional embedding vectors that represent a descriptive topic document and a news article. Similarity or distance between these embeddings can then be used directly for classification or included in a larger model as depicted in the figure. The bottom part shows that we engineered features based on the overlap of extracted entities and keywords. The combination of manually engineered features and encoding distances is then run through a simple multilayer perceptron to produce the final classification decision. In Section V we examine how well each part performs on its own and how much we gain by putting both parts together.

A. Features Based on Entity and Keyword Overlap

The bottom part of what is shown in Figure 2 requires named entities and keywords to be extracted from text documents.

Our approach to topic detection would, in principle, work with any NLP pipeline that allows making such extractions. Nevertheless, the quality of NLP affects the overall

ISSN 2395-8618

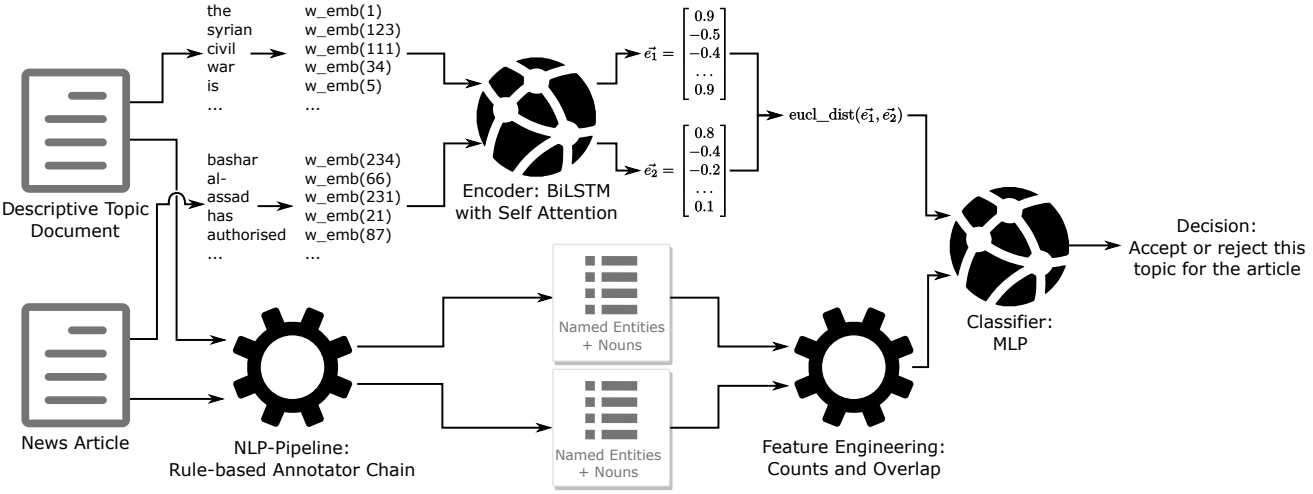

Fig. 2. A schematic view of our model. The final classification is made based on features that comprise the distance of document embeddings (top part) and manually engineered features over extracted named entities and nouns (bottom part). The LSTM and the MLP can be trained either jointly or individually.

classification result and thus we use our proprietary NLP pipeline and describe it briefly.

At first, the language is detected and the processing pipeline adapts itself to the detected language. The text is tokenized, compounds are split and a dehyphenization is done. Sentence boundaries are detected and POS tags are assigned. Lemmatization is important for extraction of proper keywords. Multi Word Expressions are detected. Named Entity Recognition (NER) is done for persons, companies, organizations and locations.

A Semantic Knowledge Graph (SKG) is used for Named Entity Identification (NEI). We build the SKG regularly using public sources (e.g., multiple Wikipedias, Wikidata, OpenStreetMap) in combination with individual sources for our clients in order to produce optimized SKGs for different use cases. Entities within the SKG have several labels, structured relationships to other entities, contextual vectors, and label-specific contexts or scores. Each piece of information found within a text document is compared to facts from the SKG in order to disambiguate multiple entities sharing the same label.

The results of our NLP pipeline are then used to extract 13 real-valued features for a pair of topic and news article: 6 features each to characterize the overlap of extracted keywords and entities and 1 feature for the cosine similarity between the average GloVe [11] embeddings over extracted keywords. For a descriptive topic document $D1$ and a news article $D2$, the six features to characterize overlap are:

1) the number of distinct items in $D1$
2) the number of distinct items in $D2$
3) the number of distinct items that occur in both
4) the sum over the *tf-idf* values for items in $D1$
5) the sum over the *tf-idf* values for items in $D2$
6) the sum over the *tf-idf* values for items that occur in both

The intention behind the *idf*-normalization is that some entities and keywords are very frequent. Just because two news documents frequently mention the *USA*, that does not mean that they are about the same thing. However, if two documents mention a rather specific entity, e.g., a fugitive terrorist, this is a much stronger signal. We compute *idf* values for identified entities and keywords over a set of 5.5 million English news articles from the year 2018 that have been processed with our NLP pipeline.

In our experiments (Section V), we show that these features together, and to a lesser extent also the three groups (average embedding, keyword overlap, entity overlap) in isolation, can be used for topic detection.

B. Text Encoder

The features presented in Section IV-A are relatively rough. With this shallow form of text understanding, the models are bound to hit a quality ceiling eventually. State-of-the-art models for text understanding (e.g., for classification or translation tasks) come with enough complexity to reach much higher ceilings. The question is how to apply them to our problem, especially due to its zero-shot nature.

We follow an approach that is inspired by FaceNet [10]. Just like FaceNet uses typical models for image processing to encode anchor and input images, we use models for NLP to encode topic descriptions and the input news articles. In FaceNet and in our work, the resulting encodings can then be compared and a pair with high-enough similarity is accepted. The top part of Figure 2 illustrates this principle for our use-case.

Descriptive documents and news articles are interpreted as sequences of word embeddings, i.e., every word of the input text is replaced by a high-dimensional vector that represents its meaning. We have experimented with (1) publicly available pre-trained word embeddings, like GloVe [11] and

ISSN 2395-8618

word2vec [12], (2) self-trained GloVe embeddings on our own corpus of news articles and (3) randomly initialized embeddings that are learned during the training of the text encoder. Differences were rather small, and in order to be flexible w.r.t. changes to lemmatization or the NLP pipeline (see Section IV-A), our experiments use randomly initialized embeddings that are learned on the fly. This also allows the model to learn an embedding for *unknown*, out-of-vocabulary words.

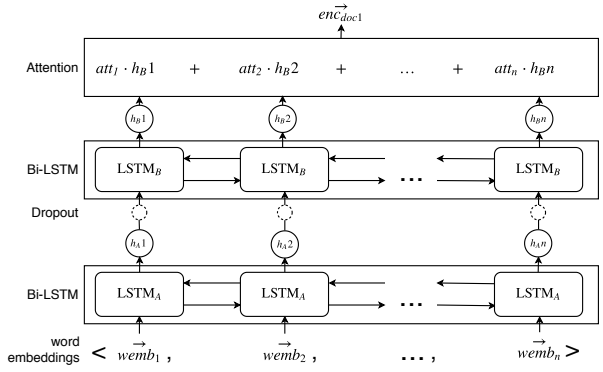

Fig. 3. Our network architecture to produce document encodings. The word embeddings pertaining to the input text are passed through two bidirectional LSTM layers. Finally, an attention layer is used to obtain the final document encoding. This layer is an implementation of the attention mechanism described in [13].

Next, the sequences of word embeddings are passed through the Neural Network depicted in Figure 3 to produce document embeddings. Finally, a pair of document encodings is compared using a simple distance function, e.g., based on euclidean distance or the cosine similarity between the two embedding vectors.

If we use the model in isolation (i.e., not as part of an ensemble), we add a fully-connected layer with one unit and sigmoid activation to find the optimal threshold to accept or reject pairs based on their distance.

We have trained these models in two ways: In the standard way, i.e., as a binary classifier that takes a pair of topic and document together with a binary true/false label, and secondly with the triplet loss that is used to train FaceNet. In our experiments using the triplet loss was not beneficial (not very harmful either) and thus we train our encoders like a standard binary classifier. This way we can use the same setup for the classifier based on overlap features, for the encoders, and for their combination. One explanation for the lack of advantages from training with the triplet loss could be that two news articles may touch the same topic but in addition to that, touch different further topics as well. News articles for the same topic can be fundamentally different from each other and are not simply variations of the same thing. In contrast, positive example images for the same face to be recognized may also be very different images, but the person depicted is the same and a perfect model might extract the same features identifying the person.

While training the model is computationally expensive, performance is not really an issue during inference. We have to compute many similarities (in particular for each news article as many as we have available topics), but the expensive computation of the encoding only has to be done once for each topic and for each news article. Hence, our topic detection scales to several thousand of topics with negligible processing times per document in comparison to the time spent in the NLP pipeline.

C. Combination

We use all features from Section IV-A and the distance of encodings from Section IV-B as input for a basic binary classifier.

We have tried several model architectures and settled for a small multilayer perceptron (MLP) with the following layers: The input features are concatenated and passed through a fully-connected layer with 10 units and ReLU activation. Then we apply batch normalization and pass the result through a fully-connected layer with 5 units and ReLU activation. Finally, we apply batch normalization again, and add a layer with one unit and sigmoid activation for prediction of the value.

We train networks that include the textual encoders, i.e., we build a joint model. We experimented with updating the encoder's weights during training and with pre-training the encoders separately and fixing their weights for training the final classifier.

V. Experiments

In the absence of an established benchmark for our problem, it is hard to provide metrics where absolute values convey much insight. Especially the available topics and their descriptive documents have a huge impact: One aspect is granularity, our Wikipedia-based heuristic extracts topics for various battles, sieges and offensives in wars (e.g., *Manbij offensive* or *Battle of Aleppo (2012–2016)*) and for rounds in sports competitions (e.g., *2018–19 UEFA Champions League knockout phase*). Assigning the correct topics out of closely related topics is much harder than distinguishing between broader topics like *Syrian Civil War* and *Soccer*.

Another aspect is the quality of descriptive documents: Some Wikipedia articles may be perfectly suited for our approach, others only contain tabular data or compile links to other articles.

A. Data

For our experiments we used data from two sources: (1) automatically retrieved positive examples for news articles about topics as described in Section III which we augment by random topic-news pairs as negative examples and (2) manually labeled topic-news pairs that were pre-selected to be *difficult*.

ISSN 2395-8618

1) Easy & Large: The process described in Section III, with picking 10 negative pairs per positive example, gives us 78,745 positive and 787,450 negative pairs about 4,390 distinct topics and 79,042 distinct news articles. We shuffled the data and reserved 10,000 pairs as a test set for evaluation. This leaves us 866,195 pairs for training.

Recall that the random process involved in creating this dataset leads to pairs where positive examples are relatively easy to distinguish from negative ones. However, since we obtain the news articles through automatic content extraction, their text may have been extracted imperfectly (or may be entirely wrong in the case of outdated links). This makes training harder and guarantees imperfect scores during evaluation.

2) Difficult & Small: Working with the automatically retrieved dataset showed the need for harder training data. Models learned to achieve very high accuracy on the reserved test set but these results did not adequately reflect the perceived quality of the models. In fact, the need for hard examples to learn from and to evaluate on is not unique to our situation: The training in [10] is also directed to prefer difficult negative examples. Unlike there, we do not have a large amount of examples to choose the difficult ones from. If we arbitrarily select semantically close topics for the news articles from the *Easy & Large* dataset, we cannot be sure that these are legitimate negative examples. After all, there may be many relevant topics for a single news article.

Thus, we selected 65 news articles from the last year at random, provided semantically close topics and then asked 17 judges to manually label them as correct or incorrect. The semantically close topics were chosen as the union of the top 10 closest topics of various early versions of our models. This strategy is reminiscent of the pooling done in various competitions, e.g., [14].

Each judge could decide how many documents she or he wanted to annotate. Judges had the possibility to skip individual topics for the article they were reviewing. We only kept documents labeled by at least 3 judges and then only kept pairs for which there was a lead (either for *correct* or for *incorrect*) of at least 2. This leaves us 61 news articles with 180 positive and 2,362 negative topic-article pairs which we split into a training set with 2,042 pairs and a reserved evaluation set with 500 pairs.

B. Results

We compare the following setups, which are described in detail in Section IV: (Avg. GloVe) a decision boundary on the cosine between the average GloVe embeddings of the news article and descriptive documents; (KW Overlap) and (Entity Overlap) the MLP using the features to express overlap; (Pairwise) the MLP using the combination of the three previous setups; (RNN Enc.) a decision boundary on the L2 distance between the document encodings produced by our recurrent neural network; (Ensemble) the MLP operating on the combination of the pairwise features and the L2 between document encodings; This ensemble was trained by using the encoder from the (RNN Encoding) setup and freezing its weights during training.

Table I depicts the performance of the models when trained only on the training set from our large, automatically-retrieved dataset. While the performance looks good on the corresponding test set, we clearly see how their performance is rather lackluster on the "hard cases", i.e., the test set from Difficult & Small.

While we need the training set from the large dataset because of its sheer size, the manually labeled hard examples are actually a lot more valuable and can vastly increase the quality of our models. Table II shows the refined results when we include the training set from our Difficult & Small data into the model training data. We do so by combining both training sets and increasing the weight of the manual examples in the loss function by a factor of 2 over the automatically generated ones.

The results show that we lose very little to no quality on the Easy & Large dataset, but increase quality a lot on Difficult & Small. We are confident that additional manually tagged examples would enable our models to achieve even better performance. To substantiate this claim, we have trained the model (Ensemble) with all automatically retrieved training examples plus 0, 100, 500, 1000 and all 2042 manually tagged training examples.

The plots in Figure 4 show how "hard examples" to learn from can vastly improve the performance on the difficult dataset, whilst losing almost no quality on the easy one. In fact, for the model (Ensemble) examined in the figure, the performance on the easy dataset yields the exact same F1 values. Tables I and II provide more details here. The plots in Figure 4 indicate that further manually tagged examples would further improve the F1-Measure because the gradient of the Difficult & Small line did not decrease yet.

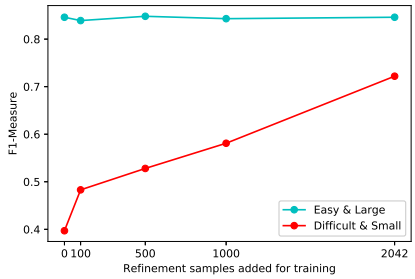

Fig. 4. Effect of adding n manually tagged examples for training the model (Ensemble).

VI. Conclusion & Future Work

We have presented and evaluated methods to detect topics in news articles by computing the similarity between the news

ISSN 2395-8618

TABLE I
RESULTS ON THE RESERVED TEST SETS FROM EITHER DATASET WHEN TRAINING ONLY WITH THE AUTOMATICALLY RETRIEVED TRAINING EXAMPLES FROM EASY & LARGE.

	Easy & Large				Difficult & Small			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
Avg. GloVe	0.923	0.879	0.134	0.232	0.928	0.5	0.111	0.111
KW Overlap	0.949	0.763	0.603	0.677	0.786	0.141	0.389	0.207
Entity Overlap	0.956	0.777	0.704	0.739	0.796	0.234	0.806	0.363
Pairwise	0.964	0.841	0.728	0.78	0.842	0.258	0.639	0.368
RNN Enc.	0.958	0.746	0.784	0.765	0.746	0.199	0.833	0.321
Ensemble	0.973	0.84	0.852	0.846	0.812	0.258	0.861	0.397

TABLE II
RESULTS ON THE RESERVED TEST SETS FROM EITHER DATASET WHEN TRAINING ON THE COMBINATION OF TRAINING EXAMPLES.

	Easy & Large				Difficult & Small			
	Acc	Prec	Rec	F1	Acc	Prec	Rec	F1
Avg. GloVe	0.921	0.872	0.111	0.197	0.934	0.8	0.111	0.195
KW Overlap	0.949	0.764	0.605	0.676	0.794	0.139	0.361	0.202
Entity Overlap	0.959	0.816	0.637	0.716	0.832	0.250	0.667	0.364
Pairwise	0.966	0.819	0.780	0.799	0.820	0.255	0.778	0.384
RNN Enc.	0.976	0.894	0.821	0.856	0.960	0.735	0.694	0.714
Ensemble	0.978	0.889	0.808	0.846	0.960	0.722	0.722	0.722

article and representative topic documents. We have shown how two kinds of input can be used to compute effective measures for similarity: (1) simple numerical features to model the overlap of mentioned entities and keywords and (2) the full texts which are passed through a recurrent neural network to produce an encoding vector. Their combination outperforms each of them individually.

While our model architecture could still be improved, there are more important aspects that should be addressed in the immediate future. Our experiments have shown that the choice of training data plays a large role. In particular, manually labeled "hard cases" are very valuable.

The more manually labeled data we have, the better results we expect. Hence, collecting more such data should be an effective way to further improve our system.

Another big issue is the choice of available topics. We have argued how this choice could make the problem very easy or arbitrarily hard. Our approach is designed so that this choice can be made per use case, possibly in a manual fashion. The current heuristic (see Section III) already turns out to be a very nice starting point and thanks to its API, Wikipedia easily allows for hourly updates to the list of available topics. If topics obtained from Wikipedia see lots of usage, some form of (automated or even manual) curation would be very helpful, e.g., elimination of obscure topics and joining cases where one logical topic is arguably split across several Wikipedia articles.

On the technical side, the search for better word embeddings is probably more urgent than further experiments with model architectures. Combining the vectors of various kinds of pre-trained embeddings, operating on character-level input, and context-sensitive embeddings are all potentially very valuable improvements. State-of-the-art systems for other NLP problems have demonstrated the value of these techniques, e.g., [15] for Named Entity Recognition. To take this idea even further, we want to experiment not only with pre-trained embeddings but entire pre-trained language models to refine and build upon. Inductive transfer learning from language models, e.g., ULMFit [16], has shown to be very powerful for multiple NLP tasks.

VII. ACKNOWLEDGMENTS

We want to thank Frank J. Balbach for a plethora of constructive comments.

REFERENCES

[1] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, Mar. 2003.

[2] T. Hofmann, "Probabilistic latent semantic indexing," in *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, ser. SIGIR '99. New York, NY, USA: ACM, 1999, pp. 50–57.

[3] D. Ramage, D. L. W. Hall, R. Nallapati, and C. D. Manning, "Labeled LDA: A supervised topic model for credit attribution in multi-labeled corpora," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing, EMNLP 2009, 6-7 August 2009, Singapore*, 2009, pp. 248–256.

[4] T. Uslu, A. Mehler, A. Niekler, and D. Baumartz, "Towards a DDC-based topic network model of Wikipedia," in *Conference: Proceedings of 2nd International Workshop on Modeling, Analysis, and Management of Social Networks and their Applications (SOCNET 2018)*, 2018.

[5] A. Joulin, E. Grave, P. Bojanowski, and T. Mikolov, "Bag of tricks for efficient text classification," *CoRR*, vol. abs/1607.01759, 2016.

[6] J. Allan, J. G. Carbonell, G. Doddington, J. Yamron, and Y. Yang, "Topic detection and tracking pilot study final report," in *Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, 1998.

ISSN 2395-8618

[7] J. Allan, R. Papka, and V. Lavrenko, "On-line new event detection and tracking," in *SIGIR*, vol. 98. Citeseer, 1998, pp. 37–45.

[8] H. Larochelle, D. Erhan, and Y. Bengio, "Zero-data learning of new tasks," in *Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence, AAAI 2008, Chicago, Illinois, USA, July 13-17, 2008*, 2008, pp. 646–651.

[9] M. Palatucci, D. Pomerleau, G. E. Hinton, and T. M. Mitchell, "Zero-shot learning with semantic output codes," in *Advances in neural information processing systems*, 2009, pp. 1410–1418.

[10] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[11] J. Pennington, R. Socher, and C. D. Manning, "GloVe: Global vectors for word representation," in *Empirical Methods in Natural Language Processing (EMNLP)*, 2014, pp. 1532–1543.

[12] 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*, 2013, pp. 3111–3119.

[13] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016, pp. 1480–1489.

[14] K. Balog, P. Serdyukov, and A. P. De Vries, "Overview of the trec 2010 entity track," Norwegian Univ of Science and Technology Trondheim, Tech. Rep., 2010.

[15] A. Akbik, D. Blythe, and R. Vollgraf, "Contextual string embeddings for sequence labeling," in *COLING 2018, 27th International Conference on Computational Linguistics*, 2018, pp. 1638–1649.

[16] J. Howard and S. Ruder, "Universal language model fine-tuning for text classification," in *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018, Melbourne, Australia, July 15-20, 2018, Volume 1: Long Papers*, 2018, pp. 328–339.