Unsupervised Keyphrase Extraction via Interpretable Neural Networks

Anonymous ACL submission

Abstract

Keyphrase extraction aims at automatically extracting a list of “important” phrases representing the key concepts in a document. Prior approaches for unsupervised keyphrase extraction resort to heuristic notions of phrase importance via embedding similarities or graph centrality, requiring extensive domain expertise to develop them. Our work presents an alternative operational definition: phrases that are most useful for predicting the topic of a text are keyphrases. To this end, we propose INSPECT—a self-explaining neural framework for identifying influential keyphrases by measuring the predictive impact of input phrases on the downstream task of topic classification. We show that this novel approach not only alleviates the need for ad-hoc heuristics but also achieves state-of-the-art results in unsupervised keyphrase extraction in 3 out of 4 diverse datasets across two domains: scientific publications and news articles. Ultimately, our study suggests a new usage of interpretable neural networks as an intrinsic component in NLP systems, and not only as a tool for explaining model predictions to humans.

1 Introduction

Keyphrase extraction is crucial for processing and understanding long documents in specialized (e.g., scientific, medical) domains (Mekala and Shang, 2020; Betti et al., 2020; Wang et al., 2019). The task is challenging, as the notion of phrase importance is context- and domain-dependent. For example, scientific terminology has key importance in processing scientific documents (Bekhuis, 2015; Gábor et al., 2016), whereas fine-grained entities and events are important in news summarization (Pighin et al., 2014; Balachandran et al., 2021; Li et al., 2016). Therefore, developing general keyphrase annotation guidelines and curating representative hand-labeled datasets is not feasible. This motivates the need for generalizable unsupervised approaches to keyphrase extraction.

So far, unsupervised approaches to keyphrase extraction have primarily relied on heuristic notions of phrase importance (Mihalcea and Tarau, 2004; Shang et al., 2018; Campos et al., 2018). Popular proxies for phrase importance include phrase clustering based on statistical features like word density (Florescu and Caragea, 2017a; Campos et al., 2018) and structural features like graph centrality (Bougouin et al., 2013). However, such approaches do not yield high-quality keyphrases in new domains as they require domain experts to carefully construct appropriate heuristics (Mani et al., 2020).

In this work, we present an alternative approach. We measure the importance of phrases in a document by their influence on classifying the topics in a text from a set of pre-defined topics. These topics are extracted unsupervisedly in a pre-processing stage. Pre-neural methods have used keyphrases to help identify topics in a document (Wallach, 2006; Wang et al., 2008; Liu and Yang, 2009). We hypothesize that end-to-end neural models also latently use keyphrases to represent documents and perform downstream tasks. Consequently, if we can interpret model decisions via highlighting salient
and influential features (phrases) used for prediction, we can identify such keyphrases. Inspired by this intuition, we propose INSPECT—a novel framework that uses interpretable text classifiers to highlight keyphrases important for predicting the topics in a text. Specifically, we adapt an interpretable classifier SelfExplain (Rajagopal et al., 2021) to jointly predict the topic of an input document and to identify the salient phrases influencing the prediction. We consider the resulting interpretations as keyphrases for the input document (§2).

Through extensive experiments, we show that INSPECT is generalizable and can be easily adapted to new domains. We present the applicability of INSPECT in the scientific and news domain (§3). Results on four benchmark datasets show that INSPECT improves keyphrase extraction performance over baselines by up to 11.4% F1 (§4), outperforming the state-of-the-art in unsupervised keyphrase extraction on 3 out 4 datasets. Importantly, INSPECT alleviates the need for heuristics and expert-labelled annotations, and thus can be applied to a wide range of domains and problems where keyphrase extraction is important. Our results confirm that the latent keyphrases obtained from an interpretable model correlate with human annotated keyphrases, opening new avenues for research on interpretable models for information extraction.1

1Code and data will be publicly released.

2 The INSPECT Framework

The goal of the INSPECT framework is to extract important keyphrases in long documents. Following a hypothesis that neural text classifiers latently leverage important keyphrases for predicting topics in text, INSPECT extracts keyphrases through interpreting the classification decisions. It builds upon an interpretable model, SelfExplain (Rajagopal et al., 2021), which learns to attribute text classification decisions to relevant phrases in the input. However, SelfExplain was designed and tested in supervised settings and for single-sentence classification; in this work we explore its extension to unsupervised keyphrase extraction from long documents. In what follows, we describe the base SelfExplain model (§2.1) and the distant supervision setup for topic classification (§2.2). We outline the training mechanism to jointly predict topics and highlight salient phrases in the document as model interpretations (§2.3) and finally extract the resulting phrase interpretations as important keyphrases in the document (§2.4). The framework overview is shown in Figure 2.

2.1 Base Interpretable Model

Feature attribution methods for model interpretability include two predominant approaches, (i) posthoc interpretations of a trained model (Jin et al., 2020; Kennedy et al., 2020; Lundberg and Lee, 2017; Ribeiro et al., 2016), and (ii) intrinsically (by-
design) interpretable models (Alvarez-Melis and Jaakkola, 2018; Rajagopal et al., 2021). We adopt the latter approach, specifically SelfExplain (Rajagopal et al., 2021) as our phrase attribution model, as the model directly produces interpretations, though in principal any phrase based interpretability techniques could be employed.

SelfExplain augments a pre-trained transformer-based model (RoBERTa (Liu et al., 2019) in our case) with a local interpretability layer (LIL) and a global interpretability layer (GIL) which are trained to produce local (relevant features from input sample) and global (relevant samples from training data) interpretations respectively. The model can be trained for any text classification tasks using gold task supervision, and produces local and global interpretations along with model predictions. Since our goal is to identify important phrases from the input sample, we use only the LIL layer. The LIL layer takes as input a sentence $x$ and a set of candidate phrases $CP = cp_1, cp_2, ..., cp_N$ and quantifies the contribution of a particular phrase for prediction through the activation difference (Shrikumar et al., 2017; Montavon et al., 2017) between the phrase and sentence representations.

### 2.2 Distant Supervision via Topic Prediction

Obtaining annotations for keyphrases in specialized domains is challenging for supervised keyphrase extraction (Mani et al., 2020). Instead, we train the interpretable model in a distant supervision setup for multi-class topic classification and use model interpretations to identify keyphrases, without any keyphrase annotations. Topical information about a document are known to be essential for identifying diverse keyphrases (Bougouin et al., 2013; Sterckx et al., 2015). Further, a comprehensive set of keyphrases should represent the various major topics in the document to be useful for different long document applications (Liu et al., 2010). We hypothesize that by using topic classification as our end-task, our model will learn to highlight—via interpretations it is designed to provide—important and diverse keyphrases in the input document.

While certain domains like news articles have extensive datasets with human annotated topic labels, others like scientific articles or legal documents require significant effort for human annotation. INSPECT can be trained using annotated topic labels when they exist. In other domains where such annotations are scarce, INSPECT can be trained using labels extracted unsupervisedly using topic models (Gallagher et al., 2017). Experiments in §4 show results using both settings.

### 2.3 Keyphrase Relevance Model

SelfExplain is designed to process single sentences and uses all the phrases spanning non-terminals in a constituency parser as units (candidate phrases) for interpretation. This is computationally expensive for our use-case. To facilitate long document topic classification, we instead define the set of noun phrases (NPs) as the interpretable units, which aligns with prior work in keyphrase extraction of using noun phrases as initial candidate phrases (Shang et al., 2018; Mihalcea and Tarau, 2004; Bougouin et al., 2013). INSPECT splits a long document into constituent passages, extracts NPs as candidates, and attributes the contribution of each NP for predicting the topics covered in the passage.

For each text block $x$ in the input document, we preprocess and identify a set of candidate phrases $CP = cp_1, cp_2, ..., cp_N$ where $N$ is the number of candidate phrases in $x$. From the base RoBERTa model, we obtain contextual [CLS] representations of the entire text block $h_{CLS}$ and individual tokens. We compute phrase representations $h_1,...,h_N$ for each candidate in $CP = \{cp_1, cp_2, ..., cp_N\}$ by taking the sum of the RoBERTa representations of each token in the phrase $cp_i$.

To compute the relevance of each phrase, we construct a representation of the input without the contribution of the phrase, $z_i$, using the activation differences between the two representations. We then pass it to a classifier layer in the local interpretability module to obtain the label distribution for prediction.

$$z_i = g(h_i) - g(h_{CLS}); \quad l_i = f(W^T z_i + b) \quad (1)$$

where $g$ is the ReLU activation function and $W$ and $b$ are the weights and bias of the classifier. Here $l_i$ denotes the label distribution obtained on passing the phrase-level representations $z_i$ through a classification layer $f$ which is either the sigmoid or the softmax function depending on the prediction task (multi-label versus multi-class). We denote the label distribution from the base RoBERTa model for predicting the output using the whole input block as $l_{CLS}$. We train the model using the cross entropy loss $\mathcal{L}_y$ with respect to the multi-label gold topics $y_i$ and an explanation specific loss $\mathcal{L}_e$ using the mean of all phrase-level label distributions such
that $l_e = \text{mean}(l_i)$.

$$
\mathcal{L}_y = -\sum_{i=1}^{T} y_i \log(l_{CLS}), \quad \mathcal{L}_e = -\sum_{i=1}^{T} y_i \log(l_e)
$$

The classifier is regularized jointly with $\alpha$ parameter using explanation and classification loss:

$$
\mathcal{L} = (1 - \alpha)\mathcal{L}_y + \alpha\mathcal{L}_e,
$$

### 2.4 Inference

During inference, for each predicted label $y \in Y$, where $Y$ denotes set of all predicted labels for input text $x$, INSPECT calculates an importance score $r_i^y$ with respect to the predicted label $y$ using the difference between the label distribution $l_i^y$ for a candidate phrase $c_i$ and the one obtained using the entire input $l_i^{CLS}$ as $r_i^y = l_i^{CLS} - l_i^y$.

This score denotes the influence of a candidate keyphrase on the predicted topic. A higher score is caused by a high shift in label distribution when using the representation of the input without the contribution of the phrase, indicating that the phrase is highly relevant for predicting the topic. Since the relevance scores are computed with respect to a particular predicted topic and it’s label distribution, the scores for the same input are not comparable across different predicted topics in multi-label classification (since label distributions can vary in magnitude). To aggregate important keyphrases across all predicted topics, we pick the ones that positively impact prediction for each topic (having a positive influence score) as a set of keyphrases.

$$
KP(x) = \{CP_i \; \forall \; r_i^y > 0; \; y \in Y; \; i \in \{1 : N\}\}
$$

### 3 Experimental Setup

#### 3.1 Evaluation Datasets

We evaluate INSPECT in two domains using four popular keyphrase extraction datasets—scientific publications (SemEval-2017 (Augenstein et al., 2017a), SciERC (Luan et al., 2018), SciREX (Jain et al., 2020)) and news articles (500N-KPCrowd (Marujo et al., 2013)). Dataset details and statistics are listed in Table 5 in the Appendix §A.1.

#### 3.2 Topic Labels

We create distant supervision for INSPECT by labeling the above datasets using document topics as labels. We leverage existing topic annotations when such annotations exist. In the 500N-KPCrowd news based dataset, we use existing topic labels (tags or categories such as Sports, Politics, Entertainment) in a one-class classification setting. For the scientific publications domain, we use topic models (Gallagher et al., 2017) to extract $T = 75$ topics where each document can be labeled with multiple topics. The scientific domain datasets are trained in a multi-label classification setup.

### 3.3 Training Data and Settings

We evaluate the generalizability of INSPECT in two experimental settings:

1. **INSPECT**: In this setting, for each of our datasets we train the model for topic prediction using only the documents and topic labels from the training set of the dataset. We then evaluate using the held-out test data from the dataset. The training data in this setting, is most closely aligned to the test data, as the documents are of the same topic distribution.

2. **INSPECT-ZeroShot**: Here, the model is trained using a large external dataset of documents and topic labels from a similar domain. The model is then evaluated on the test data of each dataset. The training data here is of a similar domain (e.g. ICLR papers for scientific domain), but not necessarily of similar topic distribution as the test data (e.g. SemEval-2017 has Physics papers which have different topics than ICLR papers). In this setting, we use data from ICLR OpenReview papers for scientific domain and BBC News articles for news domain. We collect over 8,317 full papers from ICLR and obtained 75 topic labels using topic modeling\(^2\). We removed 22 topic labels that were uninformative (list in Appendix Table 6) and used the rest to train our model in a multi-label classification setup.

    The BBC News corpus (Greene and Cunningham, 2006) consists of 2,225 news article documents, each annotated with one of five topics (business, entertainment, politics, sport, or tech).

    We pre-process each document by splitting it into text blocks of size 512 tokens, where consecutive blocks overlap with a stride size of 128. Following Shang et al. (2018), for each block we consider all Noun Phrases (NPs) as candidate phrases \(^3\).

---

\(^2\)https://openreview.net/group?id=ICLR
c\(^3\)https://github.com/gregversteeg/corex_topic
and extract them using a Noun Phrase extractor from the Berkeley Neural Parser\(^4\). All hyperparameters were chosen based on development set performance on SciERC. Our final models were trained with a batch size of 8 a learning rate of 2e-5 for 10 epochs. The classification layer dimension was 64 and \(\alpha\) was 0.5.\(^5\)

### 3.4 Baselines

We compare our method against seven unsupervised keyphrase extraction techniques — Yake (Campos et al., 2018), TF-IDF (Florescu and Caragea, 2017a), TopicRank (Bougouin et al., 2013) AutoPhrase (Shang et al., 2018; Liu et al., 2015), SifRank (Sun et al., 2020), AttentionRank (Ding and Luo, 2021) and UKE-CCRank (Liang et al., 2021). Out of the chosen baselines, Yake, TF-IDF and AutoPhrase are statistical, TopicRank is graph-based and SifRank, AttentionRank and UKE-CCRank are neural embedding based methods. Following prior work and task guidelines (Augenstein et al., 2017a; Jain et al., 2020), INSPECT produces span level keyphrases and distinguishes each occurrence of a keyphrase. In contrast, methods like SifRank, AttentionRank, and UKE-CCRank are phrase level keyphrase extractors which don’t provide span level outputs. To maintain common evaluation, we adapt these methods to span level keyphrase extraction by matching each output keyphrase to all occurrences of the phrase in the document. As our method applies a cutoff on relevance scores and picks any phrase with a positive relevance score as a keyphrase, we cannot be directly compared with baselines which rank candidate phrases and pick top-K phrases as important. To establish a fair setting for evaluation, we use the choose the average of the number of keyphrase predictions from our model as the 'K' across all baselines.

### 3.5 Evaluation Metrics

**Topic Prediction Evaluation:** To ensure high-quality interpretations from our model, it is imperative that it performs well on topic prediction. We first evaluate INSPECT’s performance on topic prediction using average F1 scores across all labels.

**Keyphrase Extraction Evaluation:** For our primary evaluation of keyphrase extraction, we evaluate using span match of our predictions and the true labels (human annotated keyphrases). Prior works (Shang et al., 2018; El-Beltagy and Rafea, 2009; Bougouin et al., 2013) have mainly focused on exact match performance. However, a recent survey highlights that the measure is highly restrictive (Papagiannopoulou and Tsoumakas, 2019) as simple variations in preprocessing can misalign phrases giving an inaccurate representation of the model’s capabilities (Boudin et al., 2016).

Alternatively, partial span match using the word level overlap between the predicted and gold span ranges, has also been explored (Rousseau and Vazirgiannis, 2015). But, it is sometimes lenient in scoring. Papagiannopoulou and Tsoumakas (2019) suggest average of the exact and partial matching as an appropriate metric based on empirical studies. Therefore, we evaluate performance using the average of the exact and partial match F1 scores between predicted and true phrases keyphrases.

### 4 Results

#### 4.1 Topic Prediction with INSPECT

First, we compare INSPECT’s effectiveness in classifying the topics with the corresponding non-interpretable encoder baseline, using micro, macro, and weighted F1 score of the classifier’s predictions compared to gold standard annotations. The results in Table 1 show that our approach outperforms a strong RoBERTa (Liu et al., 2019) baseline for topic prediction across all of our evaluation datasets. The difference is more pronounced in larger datasets (SciREX, ICLR, and BBC News), and strong performance on the topic classification task provides confidence that highlighted interpretations are for relevant and major topics in the text.

---

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>F1 Score</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Micro</td>
</tr>
<tr>
<td>SciERC</td>
<td>RoBERTa</td>
<td>0.842</td>
</tr>
<tr>
<td></td>
<td>INSPECT</td>
<td>0.836</td>
</tr>
<tr>
<td>SciREX</td>
<td>RoBERTa</td>
<td>0.609</td>
</tr>
<tr>
<td></td>
<td>INSPECT</td>
<td>0.628</td>
</tr>
<tr>
<td>SemEval17</td>
<td>RoBERTa</td>
<td>0.819</td>
</tr>
<tr>
<td></td>
<td>INSPECT</td>
<td>0.822</td>
</tr>
<tr>
<td>500N-KPCrowd</td>
<td>RoBERTa</td>
<td>0.916</td>
</tr>
<tr>
<td></td>
<td>INSPECT</td>
<td>0.938</td>
</tr>
<tr>
<td>ICLR</td>
<td>RoBERTa</td>
<td>0.729</td>
</tr>
<tr>
<td></td>
<td>INSPECT</td>
<td>0.743</td>
</tr>
<tr>
<td>BBC News</td>
<td>RoBERTa</td>
<td>0.880</td>
</tr>
<tr>
<td></td>
<td>INSPECT</td>
<td>0.902</td>
</tr>
</tbody>
</table>

Table 1: Proxy Task (Topic prediction) performance. Our INSPECT method outperforms a strong RoBERTa baseline on Micro, Macro and Weighted F1 scores.

---

\(^4\)https://pypi.org/project/benepar/

\(^5\)Details on our hyperparameter search is shared in the appendix §A.2
Table 2: Span-match results for unsupervised keyphrase extraction across datasets in the INSPECT setting. Best performance is indicated in Bold. Our model ourperforms baselines on average of exact and partial F1 scores.

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Exact Match F1</th>
<th>Partial Match F1</th>
<th>Avg Exact Partial F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SciERC</td>
<td>TF-IDF</td>
<td>0.2860</td>
<td>0.1743</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TopicRank</td>
<td>0.6580</td>
<td>0.4110</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Yake</td>
<td>0.3125</td>
<td>0.2053</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AutoPhrase</td>
<td>0.4075</td>
<td>0.2053</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AttentionRank</td>
<td>0.4075</td>
<td>0.2053</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UKE CCRank</td>
<td>0.4194</td>
<td>0.4194</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INSPECT</td>
<td>0.4316</td>
<td>0.4316</td>
<td></td>
</tr>
<tr>
<td>SciREX</td>
<td>TF-IDF</td>
<td>0.3690</td>
<td>0.2605</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TopicRank</td>
<td>0.4122</td>
<td>0.3210</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Yake</td>
<td>0.3734</td>
<td>0.2787</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AutoPhrase</td>
<td>0.4236</td>
<td>0.3025</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AttentionRank</td>
<td>0.4236</td>
<td>0.3025</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UKE CCRank</td>
<td>0.4059</td>
<td>0.3025</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INSPECT</td>
<td>0.4316</td>
<td>0.4316</td>
<td></td>
</tr>
<tr>
<td>SemEval17</td>
<td>TF-IDF</td>
<td>0.2698</td>
<td>0.1654</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TopicRank</td>
<td>0.4312</td>
<td>0.3276</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Yake</td>
<td>0.3644</td>
<td>0.2665</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AutoPhrase</td>
<td>0.404</td>
<td>0.2097</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AttentionRank</td>
<td>0.4422</td>
<td>0.2925</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UKE CCRank</td>
<td>0.345</td>
<td>0.2938</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INSPECT</td>
<td>0.5185</td>
<td>0.3899</td>
<td></td>
</tr>
<tr>
<td>500N-KPCrowd</td>
<td>TF-IDF</td>
<td>0.1703</td>
<td>0.2277</td>
<td></td>
</tr>
<tr>
<td></td>
<td>TopicRank</td>
<td>0.2346</td>
<td>0.1703</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Yake</td>
<td>0.2346</td>
<td>0.1703</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AutoPhrase</td>
<td>0.2599</td>
<td>0.2599</td>
<td></td>
</tr>
<tr>
<td></td>
<td>AttentionRank</td>
<td>0.3237</td>
<td>0.2599</td>
<td></td>
</tr>
<tr>
<td></td>
<td>UKE CCRank</td>
<td>0.203</td>
<td>0.203</td>
<td></td>
</tr>
<tr>
<td></td>
<td>INSPECT</td>
<td>0.3920</td>
<td>0.2764</td>
<td></td>
</tr>
</tbody>
</table>

4.2 Keyphrase Span Match Performance

Next, we study the utility of INSPECT in highlighting keyphrases via model interpretations. The results for INSPECT are detailed in Table 2 and, for INSPECT-ZeroShot in Table 3.

Results in Table 2 show that even with access to only training set of documents from each dataset, on 3 out of 4 datasets INSPECT outperforms all baselines with ~3.2 average F1 improvements. In the news domain (500-KPCrowd dataset) INSPECT has low exact match scores but higher partial match scores indicating misalignments between predicted and gold spans. Additionally, 500N-KPCrowd annotates all instances of a keyphrase as a reference span which favours phrase level methods like AttentionRank in the current evaluation setup. In SciREX, we observe very poor performance of UKE CCRank as it ranks common phrases like “image”, “label”, “method”, etc, very high.

In the INSPECT-ZeroShot setting, with access to a larger dataset of external documents, our model outperforms prior methods in 3 out of 4 datasets with ~3.2 points average F1 improvements. In the 500N-KPCrowd dataset, INSPECT performs comparably to SifRank with improved Partial Match F1. As Table 3 illustrates, we notice that the model consistently performs better in the INSPECT-ZeroShot setting when compared with the INSPECT setting, showing that the method benefits from more training data. Our results further show that variations in topic distribution between training and test data don’t significantly impact results. INSPECT can thus benefit from large unlabeled documents from similar domains to improve results.

INSPECT improves performance in settings with human annotated topics (news) as well as when topics are extracted using unsupervised topic modeling (scientific). Additionally, most baselines rely on carefully constructed pre- and post-processing to eliminate common phrases and produce high-quality candidates (Liang et al., 2021; Ding and Luo, 2021; Sun et al., 2020). In contrast, INSPECT achieves competitive results without domain expertise and processing for extracting quality keyphrases. Therefore, INSPECT can be easily adapted to new domains without human annotations for topics and with minimal domain knowledge.

As SifRank uses external data to augment the model, we compare it with INSPECT-ZeroShot for a fair comparison. Baselines that don’t make use of any external corpus are only included in the INSPECT setting evaluation.
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Method</th>
<th>Exact Match F1</th>
<th>Partial Match F1</th>
<th>Avg Exact Partial F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SciERC</td>
<td>TF-IDF</td>
<td>0.2162</td>
<td>0.4434</td>
<td>0.3298</td>
</tr>
<tr>
<td></td>
<td>AutoPhrase</td>
<td>0.2416</td>
<td>0.6130</td>
<td>0.4273</td>
</tr>
<tr>
<td></td>
<td>SiRank</td>
<td>0.2248</td>
<td>0.7357</td>
<td>0.4803</td>
</tr>
<tr>
<td></td>
<td>Our</td>
<td>0.4371</td>
<td>0.7114</td>
<td>0.5743</td>
</tr>
<tr>
<td>SciREX</td>
<td>TF-IDF</td>
<td>0.1780</td>
<td>0.4008</td>
<td>0.2894</td>
</tr>
<tr>
<td></td>
<td>AutoPhrase</td>
<td>0.2583</td>
<td>0.4993</td>
<td>0.3788</td>
</tr>
<tr>
<td></td>
<td>SiRank</td>
<td>0.1234</td>
<td>0.3957</td>
<td>0.2595</td>
</tr>
<tr>
<td></td>
<td>Our</td>
<td>0.2601</td>
<td>0.4893</td>
<td>0.3747</td>
</tr>
<tr>
<td>SemEval17</td>
<td>TF-IDF</td>
<td>0.1810</td>
<td>0.3398</td>
<td>0.2604</td>
</tr>
<tr>
<td></td>
<td>AutoPhrase</td>
<td>0.1104</td>
<td>0.4874</td>
<td>0.2989</td>
</tr>
<tr>
<td></td>
<td>SiRank</td>
<td>0.2804</td>
<td>0.6336</td>
<td>0.457</td>
</tr>
<tr>
<td></td>
<td>Our</td>
<td>0.3246</td>
<td>0.6218</td>
<td>0.4732</td>
</tr>
<tr>
<td>500N-KPCrowd</td>
<td>TF-IDF</td>
<td>0.1398</td>
<td>0.3578</td>
<td>0.2488</td>
</tr>
<tr>
<td></td>
<td>AutoPhrase</td>
<td>0.1701</td>
<td>0.3918</td>
<td>0.2805</td>
</tr>
<tr>
<td></td>
<td>SiRank</td>
<td>0.1847</td>
<td>0.4125</td>
<td>0.2986</td>
</tr>
<tr>
<td></td>
<td>Our</td>
<td>0.1776</td>
<td>0.4194</td>
<td>0.2985</td>
</tr>
</tbody>
</table>

Table 3: Span-match results for unsupervised keyphrase extraction in INSPECT-ZeroShot (trained on ICLR and BBC News corpus). Best performance is indicated in Bold. **INSPECT outperforms most baselines.**

Our results demonstrate that phrase attribution techniques from interpretability literature can be leveraged to identify high-quality document keyphrases by measuring predictive impact of input phrases on topic prediction. Crucially, as these keyphrases correlate with human annotated keyphrases, our results validate our initial hypothesis that neural models latently use document keyphrases for tasks like topic classification.

**5 Discussion**

Here, we present an analysis on the common error types in INSPECT and discuss the strengths and weaknesses of INSPECT using qualitative examples.

**Entity Type Analysis:** We leverage the entity type information in SciERC to observe the performance of INSPECT on specific types of keyphrases. From Table 4, we see that INSPECT performs best on keyphrases labelled as Scientific Terms and Materials. Generic phrases and Metrics are usually not representative of topical content, and thus, our method performs poorly on them. On manual analysis, we noticed that many phrases marked as Task are very unique and infrequent, making them harder to identify. A high partial match recall but a low exact match recall for Method type suggest that many predicted keyphrases are misaligned with the gold labels. We believe that alternative downstream tasks can be explored in future to help tailor our approach to capture specific types of entities, based on application requirements.

**Qualitative Analysis** In Figure 3, we show two randomly selected abstracts from the SciERC dataset. We see that INSPECT tends to extract longer phrases compared to AutoPhrase, which tends to extract mostly unigrams or bigrams. Since...
noun phrases can overlap, we observe that our model sometimes predicts overlapping phrases. Overall, our approach is able to extract more relevant phrases than the baseline. Both INSPECT and AutoPhrase tend to miss generic phrases like ‘approach’ (e.g., as seen in case 1). We hypothesize that since the downstream task in INSPECT is to identify topics, it would lead the model to focus on phrases more relevant for detecting the topic of the document. Also, this could make INSPECT miss highly targeted phrases (which usually consist of proper nouns) like Nitzan-Paroush in case 2.

INSPECT tends to extract longer compound phrases connected by functional words. Potentially, post-processing to remove overlapping and compound phrases might lead to even higher performance on datasets with smaller keyphrases. Case 2 in Figure 3 also demonstrates the ability of predicting complete phrases, like ‘classical decision-theoretic problem’, instead of AutoPhrase’s prediction – ‘classical decision-theoretic’ which is incomplete.

### 6 Related Work

Unsupervised keyphrase extraction is typically treated as a ranking problem, given a set of candidate phrases (Shang et al., 2018; Campos et al., 2018; Florescu and Caragea, 2017a). A standard pipeline (1) extracts candidate phrases; (2) scores phrase relevance; and (3) ranks the phrases based on their scores. Broadly, prior approaches can be categorized as statistical, graph-based, embedding-based, or language model based methods; Papagiannopoulou and Tsoumakas (2019) provide a detailed survey.

Statistical methods exploit notions of information theory directly. Common approaches include TF-IDF based scoring (Florescu and Caragea, 2017a) of phrases with other co-occurrence statistics to enhance performance (Liu et al., 2009; El-Beltagy and Rafea, 2009). Campos et al. (2018) shows the importance of incorporating statistical information of the context of each phrase to improve performance. Statistical approaches typically treat different instances of a phrase equally, which is a limitation.

Graph-based techniques, on the other hand, broadly aim to form a graph of candidate phrases connected based on similarity to each other. Then core components of the graph are chosen as keyphrases. Amongst these, PageRank (Brin and Page, 1998) and TextRank (Mihalcea and Tarau, 2004) assign scores to nodes based on their influence. A common extension is to use weights on the edges denoting the strength of connection (Wan and Xiao, 2008; Rose et al., 2010; Bougouin et al., 2013). Position Rank (Florescu and Caragea, 2017b) and SGRank (Danesh et al., 2015) combine the ideas from statistical, word co-occurrence and positional information. Some approaches, especially applied in the scientific document setting, make use of citation graphs (Gollapalli and Caragea, 2014; Wan and Xiao, 2008), and external knowledge bases (Yu and Ng, 2018) to improve keyphrase extraction. In this work, we focus our approach on a general unsupervised keyphrase extraction setting applicable to any domain where such external resources may not be present.

Finally, embedding based techniques (Bennani-Smires et al., 2018; Papagiannopoulou and Tsoumakas, 2018) make use of word-document similarity using word embeddings (Sun et al., 2020; Liang et al., 2021), while language-model based techniques use the uncertainty when predicting words to decide informativeness (Tomokiy and Hurst, 2003). Ding and Luo (2021) uses attention scores to calculate phrase importance with the document in an unsupervised manner.

### 7 Conclusion and Future Work

In this work, we introduced INSPECT, a novel approach to unsupervised keyphrase extraction. Our framework uses a neural model that explains text classification decisions to extract keyphrases via phrase-level feature attribution. Using four standard datasets in two domains, we show that INSPECT outperforms prior methods and establishes state-of-art results in 3 out of 4 datasets. Through qualitative and quantitative analysis, we show that INSPECT can produce high-quality and relevant keyphrases. INSPECT presents applications of interpretable models beyond explanations for humans.
References


Florian Boudin, Hugo Mougard, and Damien Cram. 2016. How document pre-processing affects keyphrase extraction performance. In NUT@COLING.


Jialu Liu, Jingbo Shang, Chi Wang, Xiang Ren, and Jiawei Han. 2015. Mining quality phrases from massive text corpora. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data, pages 1729–1744.

Nan Liu and Christopher C. Yang. 2009. Keyphrase extraction for labeling a website topic hierarchy. In ICCE.


Daniele Pighin, M. Cornolti, Enrique Alfonseca, and Katja Filippova. 2014. Modelling events through memory-based, open- ie patterns for abstractive summarization. In ACL.


10

Jingbo Shang, Jialu Liu, Meng Jiang, Xiang Ren, Clare R Voss, and Jiawei Han. 2018. Automated phrase mining from massive text corpora. IEEE Transactions on Knowledge and Data Engineering, 30(10):1825–1837.


A Appendix

A.1 Evaluation Datasets

SemEval-2017 (Augenstein et al., 2017a) consists of 500 abstracts taken from 12 AI conferences covering Computer Science, Material Science, and Physics. The entities are annotated with Process, Task, and Material labels, which form the fundamental concepts in scientific literature. Identification of the keyphrases was subtask A of the SciSemIE SemEval task (Augenstein et al., 2017b).

SciERC (Luan et al., 2018) extends SemEval-2017 by annotating more entity types, relations, and co-reference clusters to include broader coverage of general AI. The dataset was annotated by a single domain expert who had high (76.9%) agreement with three other expert annotators on 12% subset of the dataset.

SciREX (Jain et al., 2020) is a document-level information extraction dataset, covering entity identification and n-ary relation formation using salient entities. Human and automatic annotations were used to annotate 438 full papers with salient entities, with a distant supervision from the Papers With Code7 corpus. This dataset can help verify the performance of models on full papers.

500N-KPCrowd (Marujo et al., 2013) is a keyphrase extraction dataset in the news domain. This data consists of 500 articles from 10 topics annotated by multiple Amazon Mechanical Turk workers for important keywords. Following the baselines on this datasets, we pick keywords that were among the top two most frequently chosen by the human annotators. Since no span-level information for these keywords is given, we annotate all occurrences of the chosen keywords in the document to obtain a list of span labels, which we use to evaluate all the models.

A.2 Implementation Details

Here, we present the hyper-parameters for all experiments along with their corresponding search space. We chose all hyperparameters based on the development set performance on the SciERC dataset. We considered RoBERTa (Liu et al., 2019) and XL-NET (Yang et al., 2019) based encoders and finally chose RoBERTa for faster compute times.

7https://paperswithcode.com/
<table>
<thead>
<tr>
<th>Dataset</th>
<th>Type</th>
<th>Split</th>
<th>Total docs</th>
<th>Avg words per doc</th>
<th>Avg keyphrases per doc</th>
</tr>
</thead>
<tbody>
<tr>
<td>SciERC</td>
<td>Scientific</td>
<td>Train</td>
<td>350</td>
<td>130</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dev</td>
<td>50</td>
<td>130</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test</td>
<td>100</td>
<td>134</td>
<td>17</td>
</tr>
<tr>
<td>SciREX</td>
<td>Scientific</td>
<td>Train</td>
<td>306</td>
<td>5601</td>
<td>353</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dev</td>
<td>66</td>
<td>5484</td>
<td>354</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test</td>
<td>66</td>
<td>6231</td>
<td>387</td>
</tr>
<tr>
<td>SemEval17</td>
<td>Scientific</td>
<td>Train</td>
<td>350</td>
<td>160</td>
<td>21</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dev</td>
<td>50</td>
<td>193</td>
<td>27</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test</td>
<td>100</td>
<td>186</td>
<td>23</td>
</tr>
<tr>
<td>500N-KPCrowd</td>
<td>News</td>
<td>Train</td>
<td>400</td>
<td>430</td>
<td>193</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Dev</td>
<td>50</td>
<td>465</td>
<td>86</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Test</td>
<td>50</td>
<td>420</td>
<td>116</td>
</tr>
<tr>
<td>BBC News</td>
<td>News</td>
<td>All</td>
<td>2225</td>
<td>385</td>
<td>-</td>
</tr>
<tr>
<td>ICLR</td>
<td>Scientific</td>
<td>All</td>
<td>8317</td>
<td>6505</td>
<td>-</td>
</tr>
</tbody>
</table>

Table 5: Description about the datasets. Average words and keyphrases per document are rounded to the nearest whole number. ICLR and BBC News are used in INSPECT-ZeroShot setting for training and don’t have any labelled keyphrase data.

S.No. | Top words from removed topic
--- | ---
1 | proposed;propose novel;propose;proposed method;method
2 | generalization;study;analysis;suggest;provide
3 | outperforms;existing;existing methods;outperforms stateoftheart;methods
4 | state;art;state art;shorter term;current state
5 | effectiveness;demonstrate effectiveness;source;effectiveness proposed;student
6 | training;training data;training set;training process;model training
7 | experimental;experimental results;results;results demonstrate;experimental results demonstrate
8 | experiments;extensive;extensive experiments;experiments demonstrate;conduct
9 | performance;improves;significantly;improve;improved
10 | recent;shown;recent work;recent advances;success
11 | achieves;introduce;competitive;achieves stateoftheart;introduce new
12 | trained;model trained;models trained;networks trained;trained using
13 | present;pap er present;present novel;work present;monte
14 | widely;parameters;widely used;proposes;paper proposes
15 | simple;benchmark datasets;benchmark;propose simple;simple effective
16 | prior;approach;sampling;continuous;prior work
17 | program;introduces;programs;future;paper introduces
18 | solve;challenging;able;complex;challenging problem
19 | challenge;current;challenges;open;current stateoftheart
20 | rate;good;good performance;1;regime
21 | works;previous works;existing works;focus;scenarios
22 | evaluate;evaluation;tackle;tackle problem;evaluate method

Table 6: 22 Generic topics removed from the 75 topic labels learned using topic modeling on ICLR data.

We experimented with learning-rates from the set of 1e-5,2e-5,5e-5,1e-4 and 2e-4. We chose 2e-5 as the final learning rate. Our batch size of 8 was chosen after experimenting with 4, 8, 12 and 16.

The size of the weights matrix in the classification layer was chosen to be 64 from a set of 16,32,64 and 128. The $\alpha$ parameter used for regularization was fixed at 0.5. We tried values between 0.1 and 0.9 and did not find significant difference. We saved the model based on best weighted F1 on the topic prediction task. All training runs took less than 3 hours on 2 Nvidia 2080Ti GPUs, except on the ICLR dataset, which took 8 hours. All results are from a single run.