Unsupervised Keyphrase Extraction via Interpretable Neural Networks

Anonymous ACL submission

Abstract

Keyphrase extraction aims at automatically extracting a list of "important" phrases which represent the key concepts in a document. Traditionally, it has been approached from an information-theoretic angle using phrase co-occurrence statistics. This work proposes a novel unsupervised approach to keyphrase extraction that uses a more intuitive notion of phrase importance, inspired by interpretability research. In particular, we use a self-explaining neural model to measure the predictive impact of input phrases on downstream task performance, and consider the resulting interpretations as document keyphrases for the target task. We show the efficacy of our approach on four datasets in two domains—scientific publications and news articles—attaining state-of-the-art results in unsupervised keyphrase extraction.

1 Introduction

Keyphrase extraction is a crucial step in processing long documents, especially in specialized (e.g., scientific, medical) domains (Mekala and Shang, 2020; Dong et al., 2020; Betti et al., 2020; Wang et al., 2019). Identifying important keyphrases is challenging, since the notion of importance is context- and task-dependent. For example, scientific terminology has key importance in summarization of scientific documents (Bekhuis, 2015; Gábor et al., 2016), whereas fine-grained entities and events are generally important in news summarization (Pighin et al., 2014; Balachandran et al., 2021; Yang et al., 2020; Li et al., 2016). Consequently, developing general keyphrase annotation guidelines and curating hand-labeled datasets is expensive, and is not easily transferable across domains (Mani et al., 2020).

Prior approaches primarily relied on information theory to quantify phrase importance (Mihalcea and Tarau, 2004), and heuristic scoring techniques incorporating various frequency, position, and syntactic features to rank extracted phrases (Shang et al., 2018). Neural unsupervised approaches are limited to using language model scores for phrase ranking (Tomokiyo and Hurst, 2003). These prior approaches cannot be easily adapted to obtain high-quality task-specific keyphrases.

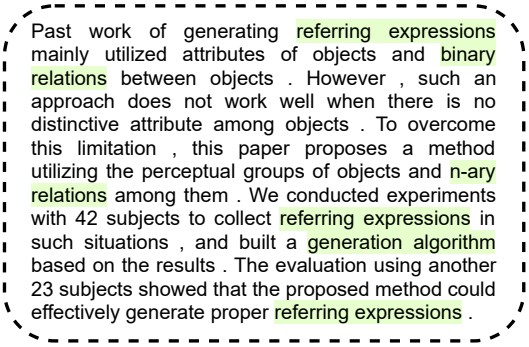

Figure 1: Example extracted keyphrases from an abstract in SciERC (Luan et al., 2018). Our task is to identify these keyphrases in an unsupervised setting.

We propose a novel neural approach to keyphrase extraction. Specifically, we leverage *importance attribution* techniques from interpretability literature in NLP (Jin et al., 2020; Kennedy et al., 2020), and a classification model—SelfExplain (Rajagopal et al., 2021)—which is interpretable by design: it learns to attribute text classification decisions to relevant phrases in the input text and in the training corpus. We adapt the SelfExplain model to process long documents, and propose a distant supervision setup to facilitate keyphrase extraction (§2). Specifically, we train the model on multi-label topic classification, and extract resulting model interpretations as important keyphrases for the document. We hypothesize that in classification of these domain-specific topics SelfExplain will learn to highlight—via interpretations it is designed to provide—important keyphrases in the input document. We call this novel framework for

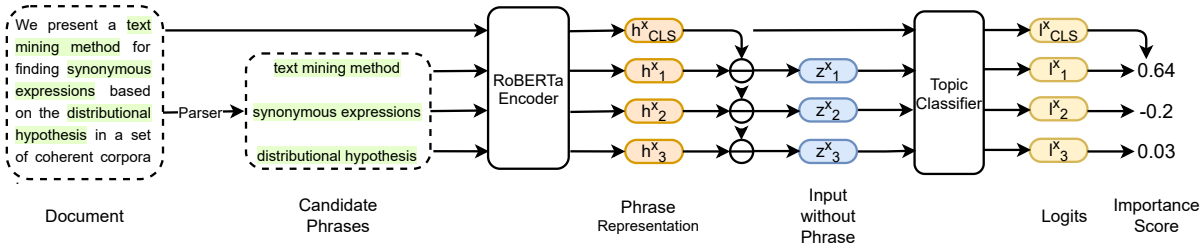

Figure 2: Overview of INSPECT. We first extract candidate phrases (all Noun Phrases) from the document using a parser before obtaining the representation of each phrase using RoBERTa. We construct representations of the input without the contribution of each of the phrases, which are then provided to our topic classifier. The difference in predictions gives us importance scores for each phrase, where a higher score signifies more influence on prediction.

interpretable **u**nsupervi**s**ed **p**hrase **e**xtra**c**tion INSPECT.

We evaluate INSPECT in two domains—scientific publications and news articles (§3.1). Results in §4 on four benchmark datasets show that INSPECT improves keyphrase extraction performance over all baselines on all datasets by up to 15% F1. The increase in performance is more pronounced in smaller datasets where frequency based methods especially struggle.

In summary, this paper's key contribution is a neural network-based framework to quantify the importance of phrases in long documents by training a self-explainable classifier on the downstream task of topic prediction. Through empirical analysis on four datasets, we show that INSPECT outperforms state-of-the-art approaches to keyphrase extraction. Importantly, INSPECT alleviates the need to collecting expert-labelled annotations and thus can be applied to a wide range of domains and problems where keyphrase extraction is important.[1]

2 The INSPECT Framework

Our INSPECT framework leverages model explanations, produced by neural interpretability approaches, to extract important keyphrases in the long documents. While we rely on an existing interpretable model to extract the phrases, the overall application of this model to inducing document structure via phrase importance scoring is novel. INSPECT identifies and scores relevant phrases in the document through the use of distant supervision in the downstream task of predicting the topic of a document. In what follows, we outline the base model, a mechanism for attributing phrase relevance to the downstream task of topic prediction, and finally scoring the candidate keyphrases in target documents. The framework overview is shown in Figure 2.

2.1 Base Model: SelfExplain

Feature attribution methods for model interpretability include two predominant approaches, (i) post-hoc explanations of a trained model (Jin et al., 2020; Kennedy et al., 2020; Lundberg and Lee, 2017), and (ii) intrinsically (by-design) explainable 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. SelfExplain augments a pre-trained transformer-based model (RoBERTa (Liu et al., 2019) in our case) with a local interpretability layer and a global interpretability layer which are trained to produce local (relevant features from input sample) and global (relevant samples from training data) explanations jointly with the model predictions. Since our goal is to identify important phrases from the input sample, we use only the local explanation layer and adapt it for topic prediction.

The local interpretability layer takes as input a sentence x and a set of candidate phrases $CP^x = cp_1^x, cp_2^x, ..., cp_N^x$ 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

SelfExplain is designed to process single sentences and uses a set of all phrases spanning non-terminals in a constituency parser as units 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

[1] Code and data will be publicly released.

keyphrase extraction (Shang et al., 2018; Mihalcea and Tarau, 2004; Bougouin et al., 2013). INSPECT splits a long document into constituent passages, extracts NPs as candidate phrases, and uses the SelfExplain model architecture to attribute the contribution of each noun phrase for topic prediction. We discuss identification of relevant NPs for topic prediction in the INSPECT framework below.

2.3 Keyphrase Relevance Model

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

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

$$z_i^x = g(h_i^x) - g(h_{CLS}^x)$$

$$l_i^x = f(W^T z_i + b)$$

where g is the ReLU activation function and W and b are the weights and bias of the classifier. Here l_i^x denotes the label distribution obtained on passing the phrase-level representations z_i^x 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}^x. We train the model using the cross entropy loss with respect to the gold topic label y_t as follows :

$$\mathcal{L}_y = -\sum_{t=1}^{T} y_t \log(l_{CLS}^x)$$

The classifier is regularized with an explanation specific loss by computing a weighted average over all the phrase-level label distributions such that $l_e = \sum_i w_i \times l_i^x$:

$$\mathcal{L}_e = -\sum_{t=1}^{T} y_t \log(l_e)$$

and computes a joint explanation and classification loss for the model as:

$$\mathcal{L} = \mathcal{L}_y + \alpha \mathcal{L}_e,$$

where α is the regularization parameter.

2.4 Inference

During inference, INSPECT calculates an importance score r_i^x using the difference between the label distribution l_i^x for the candidate phrase c_i^x and the one obtained from the entire input l_{CLS}^x as

$$r_i^x = l_{CLS}^x - l_i^x.$$

This score denotes the influence of a candidate keyphrase on the topic prediction. 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 prediction. 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.

3 Experimental Setup

3.1 Evaluation Datasets

We evaluate INSPECT in two domains—scientific publications and news articles—and on four popular keyphrase extraction datasets: SemEval-2017, SciERC, SciREX (Scientific) and 500N-KPCrowd (News). Dataset statistics are listed in Table 5 in the Appendix.

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 ScienceIE 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 Code[2] 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.

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. For example, news articles are often categorized into topics (tags or categories such as Sports, Politics, Entertainment). For the scientific publications domain, we use topic models (Gallagher et al., 2017) to extract T topics where each document can be labeled with multiple topics. For the news domain, our topic prediction task is a one-class classification problem, while for the scientific domain, it is as multi-label classification setup.

3.3 Training Data and Settings

We evaluate the generalizability of INSPECT in two experimental settings:

1. **INSPECT**: For each dataset (SciERC, SciREX, Semeval-2017 and 500N-KPCrowd), we train the model for topic prediction using only the documents in the training set of the dataset and their corresponding topic labels (obtained using the approach outlined in §3.2). The training data in this setting, is most closely aligned to the test data, where the documents are of a similar topic distrubution. We then evaluate the model on the held-out test data from the dataset.

2. **INSPECT-ZeroShot** Here, the model is trained using a large set similar-domain external dataset of documents and corresponding topic labels and 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 is not necessarily of similar topic distribution as the test data (e.g. SemEval-2017 has Physicis papers which might have different topics when compared to ICLR papers). In this setting, we use data from ICLR (OpenReview[3]) papers for scientific domain and BBC News articles for news domain to train the model on topic prediction. We collect over 8,317 full papers from ICLR and obtained 75 topic labels using topic modeling[4]. We manually removed 22 topic labels that were generic and 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. For each block, we then extract candidate phrases. Following Shang et al. (2018), we consider all Noun Phrases (NPs) as candidate phrases and extract them using a Noun Phrase extractor from the Berkeley Neural Parser[5].

We chose all hyperparameters based on the development set performance on the SciERC dataset.[6] Our final models were trained with a batch size of 8 and a learning rate of 2e-5. Our classification layer weight dimension is 64. The λ parameter used to combine the phrase and context representations was fixed at 0.5. We train each of our models

[2] https://paperswithcode.com/

[3] https://openreview.net/group?id=ICLR.cc

[4] https://github.com/gregversteeg/corex_topic

[5] https://pypi.org/project/benepar/

[6] Details on our hyperparameter search is shared in the appendix.

for 10 epochs and save the model based on best weighted F1 performance 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.

3.4 Baselines

We compare our method against four common unsupervised keyphrase extraction techniques — Yake (Campos et al., 2018), TF-IDF (Florescu and Caragea, 2017a), TopicRank (Bougouin et al., 2013), and AutoPhrase (Shang et al., 2018; Liu et al., 2015). Out of the four chosen baselines, Yake, TF-IDF and AutoPhrase are statistical, whereas TopicRank is graph-based. Yake and TopicRank are single document keyphrase extraction techniques and do not rely on additional data from external corpora to improve performance. 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 the most challenging and fair setting for evaluation, for each baseline we choose a 'K' value which gives best F1 performance in the development set.

3.5 Evaluation Metrics

Topic Prediction Evaluation: We first evaluate INSPECT's performance on the downstream proxy task of topic prediction. To ensure high-quality explanations from our model, it is imperative that it performs well on the topic prediction task. For all experiments, we evaluate 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 (keyphrases). Prior works (Shang et al., 2018; El-Beltagy and Rafea, 2009; Bougouin et al., 2013) have mainly focused on *exact match* performance, however, more recent surveys highlight issues with exact match, as the measure is highly restrictive (Papagiannopoulou and Tsoumakas, 2019). While exact span match gives high scores for exact phrases being returned by the model, it is dependent on the candidate extraction steps as simple differences in preprocessing can misalign phrases giving an inaccurate representation of the model's capabilities.

Alternatively, *partial span match* has also been explored (Rousseau and Vazirgiannis, 2015). But, it be over lenient in scoring predicted phrases. Papagiannopoulou and Tsoumakas (2019) suggest *average of the exact and partial matching* as an appropriate metric based on emperical studies. Therefore, in this work, we evaluate performance based on the average of the exact and partial match F1 scores between the predicted phrases and the gold standard keyphrases. We calculate partial match F1 scores by considering the word level overlap between the predicted and gold span ranges.

Dataset	Method	F1 Score Micro	Macro	Weighted
SciERC	RoBERTa	0.842	0.651	0.767
	INSPECT	0.836	0.658	**0.771**
SciREX	RoBERTa	0.609	0.404	0.641
	INSPECT	0.628	0.442	**0.697**
SemEval17	RoBERTa	0.819	0.613	0.731
	INSPECT	0.822	0.611	**0.744**
500N-KPCrowd	RoBERTa	0.916	0.880	0.910
	INSPECT	0.938	0.904	**0.939**
ICLR	RoBERTa	0.729	0.456	0.699
	INSPECT	0.743	0.492	**0.733**
BBC News	RoBERTa	0.880	0.851	0.876
	INSPECT	0.902	0.886	**0.894**

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

4 Results

In this section, we evaluate the performance of our proposed model compared to baselines on (1) our proxy task of topic prediction, and (2) span match performance for keyphrases extracted.

4.1 Topic Prediction with INSPECT

First, we compare INSPECT's effectiveness in predicting topics while learning phrase-level importance with an 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 proxy task supports the hypothesis that the model extracts relevant phrase explanations for model predictions.

4.2 Keyphrase Span Match Performance

Next, we compare the utility of INSPECT in extracting keyphrases against all baselines on the four

Dataset	Method	Exact Match F1	Partial Match F1	Avg Exact Partial F1
SciERC	TF-IDF	0.0627	0.2860	0.1743
	TopicRank	0.2533	**0.5680**	0.4110
	Yake	0.2230	0.5125	0.3678
	AutoPhrase	0.0961	0.3145	0.2053
	INSPECT	**0.3108**	0.5524	**0.4316**
SciREX	TF-IDF	0.1521	0.3690	0.2605
	TopicRank	0.2298	0.4122	0.3210
	Yake	0.1840	0.3734	0.2787
	AutoPhrase	0.1814	**0.4236**	0.3025
	INSPECT	**0.2397**	0.4127	**0.3262**
SemEval17	TF-IDF	0.0610	0.2698	0.1654
	TopicRank	0.2240	0.4312	0.3276
	Yake	0.1687	0.3644	0.2665
	AutoPhrase	0.0790	0.3404	0.2097
	INSPECT	**0.2594**	**0.5185**	**0.3889**
500N-KPCrowd	TF-IDF	0.1034	0.3520	0.2277
	TopicRank	0.1060	0.2346	0.1703
	Yake	0.1380	0.3551	0.2465
	AutoPhrase	0.1590	0.3608	0.2599
	INSPECT	**0.1608**	**0.3920**	**0.2764**

Table 2: Span-match results for unsupervised keyphrase extraction across datasets in the INSPECT setting. Best performance is indicated in Bold. Our model ourperforms all baselines on average of exact and partial F1 scores.

Dataset	Method	Exact Match F1	Partial Match F1	Avg Exact Partial F1
SciERC	TF-IDF	0.2162	0.4434	0.3298
	AutoPhrase	0.2416	0.6130	0.4273
	Our	**0.4227**	**0.6929**	**0.5578**
SciREX	TF-IDF	0.1780	0.4008	0.2894
	AutoPhrase	0.2583	**0.4993**	**0.3788**
	Our	**0.2601**	0.4893	0.3747
SemEval17	TF-IDF	0.1810	0.3398	0.2604
	AutoPhrase	0.1104	0.4874	0.2989
	Our	**0.3246**	**0.6218**	**0.4732**
500N-KPCrowd	TF-IDF	0.1398	0.3578	0.2488
	AutoPhrase	0.1701	0.3918	0.2805
	Our	**0.1776**	**0.4194**	**0.2985**

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 is better or comparable to other approaches.

evaluation datasets using span match evaluations. The results for INSPECT are detailed in Table 2 and, for INSPECT-ZeroShot in Table 3. All baselines in Table 2 were trained using only training documents of the corresponding dataset. Note that Yake and Topic Rank do not make use of any external corpus to learn how to predict keyphrases. For a fair comparison, we thus include them only in the INSPECT setting evaluation.

Results in Table 2 show that even with access to a small training set of documents from each dataset, INSPECT outperforms all baselines with $\sim$2.75 average F1 improvements and establishes state-of-art results on unsupervised keyphrase extraction. Even in the restrictive exact span match metric, INSPECT has an improvement between 1-6 F1 points over previous state-of-art methods. Interestingly, we observe poor performance from TF-IDF and AutoPhrase on SciERC and SemEval17 datasets. We hypothesize that in small datasets like SciERC and SemEval17, it is harder to obtain accurate statistical measures which both these methods rely on. In larger datasets (SciREX and 500N-KPCrowd), both methods improve resulting in similar performance to other baselines.

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 $\sim$10.4 points average F1 improvements. As Table 3 illustrates, we notice that the model consistently perform 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 do not significantly impact results.

Our results demonstrate that our approach of using phrase attribution based techniques to iden-

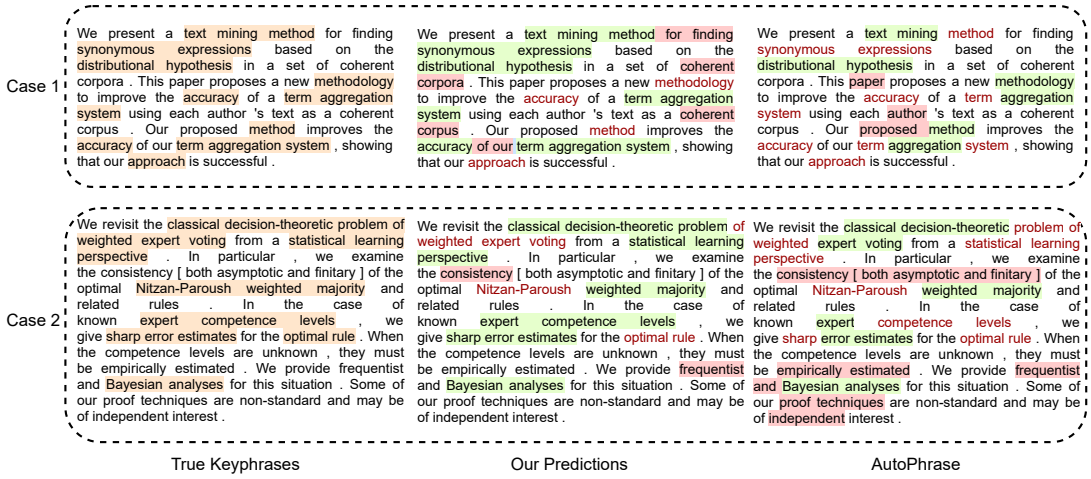

Figure 3: Two data points randomly chosen from the SciERC dataset. Orange spans represent gold standard annotations. Green spans in the predictions represent correctly predicted spans, whereas red spans are spans wrongly predicted as being keyphrases and red text are keyphrases that the model did not identify.

tify phrases with high predictive impact on a task like topic prediction can output high-quality keyphrases. This helps us introduce a new direction for keyphrase extraction.

5 Discussion

In this section, we discuss and provide examples of INSPECT's performance, strengths, and weaknesses on keyphrase extraction. We also discuss some common types of errors that our system makes and demonstrate the specific types of keyphrases that our model is better at extracting.

Entity Type Analysis: We leverage the entity type information present 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 representatives of the topical content, and thus, our method performs poorly on them. On manual inspection, we noticed that many phrases that were marked as *Task* are very specific, which might make them harder to learn. A high partial match recall but a low exact match recall for the *Method* type suggests that many predicted keyphrases are misaligned with the gold labels by a few words. We believe that using different downstream tasks can help tailor our approach to capture specific types of entities better, based on the requirement of the application that builds upon keyphrase extraction.

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). This might be due to topic prediction training incorporated as part of downstream task in INSPECT, which would lead the model to focus on phrases more relevant for detecting the topic of the document. Also, because of this, INSPECT might miss too specific phrases (which usually consist of proper nouns) like *Nitzan-Paroush* in case 2.

Due to the nature of extracting longer phrases, INSPECT extracts more compound phrases connected by functional words. We hypothesize that post-processing, to remove overlapping and compound phrases might lead to even higher performance on datasets consisting of smaller phrases. Case 2 in Figure 3 also demonstrates the trend 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 can-

Type	Recall	
	Exact	Partial
Metric	60.65	78.34
Task	58.27	90.45
Material	72.17	86.69
Scientific Term	78.87	95.13
Method	65.31	95.41
Generic	63.16	86.06

Table 4: Exact and partial span match recall scores for different types of keyphrases on the SciERC dataset.

didate 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 (Florescu and Caragea, 2017a; El-Beltagy and Rafea, 2009; Liu et al., 2009; Campos et al., 2018), graph-based (Brin and Page, 1998; Mihalcea and Tarau, 2004; Wan and Xiao, 2008; Rose et al., 2010; Danesh et al., 2015; Florescu and Caragea, 2017b; Gollapalli and Caragea, 2014; Bougouin et al., 2013; Yu and Ng, 2018), embedding-based (Bennani-Smires et al., 2018; Papagiannopoulou and Tsoumakas, 2018), or language model based methods (Tomokiyo and Hurst, 2003); Papagiannopoulou and Tsoumakas (2019) provide a detailed survey.

Statistical techniques for keyphrase extraction exploit notions of information theory directly. The most common (and surprisingly strong) baseline is TF-IDF based scoring of phrases (Florescu and Caragea, 2017a). Other approaches use phrase position in the document (El-Beltagy and Rafea, 2009) or co-occurrence statistics and semantic relatedness of candidate terms to cluster phrases (Liu et al., 2009). Capturing the statistical information of the context of each phrase has also been shown to be an important signal for keyphrase extraction (Campos et al., 2018). Statistical approaches typically treat different instances or uses 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 key phrases. Amongst these, PageRank (Brin and Page, 1998) gives the score to each node based on recursive node influence. TextRank (Mihalcea and Tarau, 2004) specifically applied this idea by connecting nodes based on co-occurrence within some window. A common extension to such techniques 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 these external resources are not present.

Finally, embedding based techniques (Bennani-Smires et al., 2018; Papagiannopoulou and Tsoumakas, 2018) make use of word-document similarity using word embeddings, while language-model based techniques use the uncertainty when predicting words to decide informativeness (Tomokiyo and Hurst, 2003).

7 Conclusion and Future Work

We propose INSPECT, a novel approach to unsupervised keyphrase extraction. Our framework uses a neural model that interprets text classification decisions to extract keyphrases via phrase-level feature attribution. Using four standard datasets, we show that INSPECT outperforms prior methods and establishes new state-of-the-art results in unsupervised keyphrase extraction. Through qualitative and quantitative analysis, we show that INSPECT can leverage large external corpora to produce high-quality keyphrases in the scientific and news domains. INSPECT also opens doors for more control in keyphrase extraction and model explanation applications. For instance, depending on the proxy task, our focus and the definition of importance can be varied. While topic prediction may be a good task to capture content, sentiment prediction might improve pragmatic understanding.

Ultimately, our work utilizes model explanations in an automated (rather that human–computer interaction) setting. With the advances in explainable and interpretable NLP, such a framework relying on feature-level attribution and model explanations to improve a downstream task, can be applied in many applications, including for unsupervised information extraction, content planning, and structured prediction.

References

David Alvarez-Melis and Tommi S Jaakkola. 2018. Towards robust interpretability with self-explaining neural networks. *Neurips*.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017a. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017b. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Vidhisha Balachandran, Artidoro Pagnoni, Jay Yoon Lee, Dheeraj Rajagopal, J. Carbonell, and Yulia Tsvetkov. 2021. Structsum: Summarization via structured representations. In *EACL*.

Tanja Bekhuis. 2015. Keywords, discoverability, and impact. *Journal of the Medical Library Association : JMLA*, 103 3:119–20.

Kamil Bennani-Smires, Claudiu Musat, Andreea Hossmann, Michael Baeriswyl, and Martin Jaggi. 2018. Simple unsupervised keyphrase extraction using sentence embeddings. *arXiv preprint arXiv:1801.04470*.

Arianna Betti, Martin Reynaert, Thijs Ossenkoppele, Yvette Oortwijn, Andrew Salway, and Jelke Bloem. 2020. Expert concept-modeling ground truth construction for word embeddings evaluation in concept-focused domains. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6690–6702, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Adrien Bougouin, Florian Boudin, and Béatrice Daille. 2013. TopicRank: Graph-based topic ranking for keyphrase extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 543–551, Nagoya, Japan. Asian Federation of Natural Language Processing.

Sergey Brin and Lawrence Page. 1998. The anatomy of a large-scale hypertextual web search engine. *Computer Networks*, 30:107–117.

Ricardo Campos, Vítor Mangaravite, Arian Pasquali, Alípio Mário Jorge, Célia Nunes, and Adam Jatowt. 2018. A text feature based automatic keyword extraction method for single documents. In *European conference on information retrieval*, pages 684–691. Springer.

Soheil Danesh, Tamara Sumner, and James H. Martin. 2015. SGRank: Combining statistical and graphical methods to improve the state of the art in unsupervised keyphrase extraction. In *Proceedings of the Fourth Joint Conference on Lexical and Computational Semantics*, pages 117–126, Denver, Colorado. Association for Computational Linguistics.

Xin Luna Dong, Xiang He, Andrey Kan, Xian Li, Yan Liang, Jun Ma, Yifan Ethan Xu, Chenwei Zhang, Tong Zhao, Gabriel Blanco Saldana, Saurabh Deshpande, Alexandre Michetti Manduca, Jay Ren, Surender Pal Singh, Fan Xiao, Haw-Shiuan Chang, Giannis Karamanolakis, Yuning Mao, Yaqing Wang, Christos Faloutsos, Andrew McCallum, and Jiawei Han. 2020. Autoknow: Self-driving knowledge collection for products of thousands of types. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '20, page 2724–2734, New York, NY, USA. Association for Computing Machinery.

Samhaa R. El-Beltagy and Ahmed Rafea. 2009. Kp-miner: A keyphrase extraction system for english and arabic documents. *Information Systems*, 34(1):132–144.

Corina Florescu and Cornelia Caragea. 2017a. A new scheme for scoring phrases in unsupervised keyphrase extraction. In *European Conference on Information Retrieval*, pages 477–483. Springer.

Corina Florescu and Cornelia Caragea. 2017b. Positionrank: An unsupervised approach to keyphrase extraction from scholarly documents. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1105–1115.

K. Gábor, Haïfa Zargayouna, D. Buscaldi, I. Tellier, and Thierry Charnois. 2016. Semantic annotation of the acl anthology corpus for the automatic analysis of scientific literature. In *LREC*.

Ryan J Gallagher, Kyle Reing, David Kale, and Greg Ver Steeg. 2017. Anchored correlation explanation: Topic modeling with minimal domain knowledge. *Transactions of the Association for Computational Linguistics*, 5:529–542.

Sujatha Das Gollapalli and Cornelia Caragea. 2014. Extracting keyphrases from research papers using citation networks. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1629–1635. AAAI Press.

Derek Greene and Pádraig Cunningham. 2006. Practical solutions to the problem of diagonal dominance in kernel document clustering. In *Proc. 23rd International Conference on Machine learning (ICML'06)*, pages 377–384. ACM Press.

Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020. Scirex: A challenge dataset for document-level information extraction.

In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*.

Xisen Jin, Zhongyu Wei, Junyi Du, Xiangyang Xue, and Xiang Ren. 2020. Towards hierarchical importance attribution: Explaining compositional semantics for neural sequence models. In *International Conference on Learning Representations*.

Brendan Kennedy, Xisen Jin, Aida Mostafazadeh Davani, Morteza Dehghani, and Xiang Ren. 2020. Contextualizing hate speech classifiers with post-hoc explanation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5435–5442, Online. Association for Computational Linguistics.

Wei Li, Lei He, and Hai Zhuge. 2016. Abstractive news summarization based on event semantic link network. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 236–246, Osaka, Japan. The COLING 2016 Organizing Committee.

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.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 conference on empirical methods in natural language processing*, pages 257–266.

Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. 2018. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium. Association for Computational Linguistics.

Scott M. Lundberg and Su-In Lee. 2017. A unified approach to interpreting model predictions. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 4768–4777, Red Hook, NY, USA. Curran Associates Inc.

Kaushik Mani, Xiang Yue, Bernal Jimenez Gutierrez, Yungui Huang, Simon Lin, and Huan Sun. 2020. Clinical phrase mining with language models. In *2020 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 1087–1090. IEEE.

Luis Marujo, Márcio Viveiros, and João Paulo da Silva Neto. 2013. Keyphrase cloud generation of broadcast news. In *Proceeding of Interspeech 2011: 12th Annual Conference of the International Speech Communication Association*.

Dheeraj Mekala and Jingbo Shang. 2020. Contextualized weak supervision for text classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 323–333, Online. Association for Computational Linguistics.

Rada Mihalcea and Paul Tarau. 2004. Textrank: Bringing order into text. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 404–411.

Grégoire Montavon, Sebastian Lapuschkin, Alexander Binder, Wojciech Samek, and Klaus-Robert Müller. 2017. Explaining nonlinear classification decisions with deep taylor decomposition. *Pattern Recognition*, 65:211–222.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2018. Local word vectors guiding keyphrase extraction. *Information Processing & Management*, 54(6):888–902.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2019. A review of keyphrase extraction. *CoRR*, abs/1905.05044.

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

Dheeraj Rajagopal, Vidhisha Balachandran, E. Hovy, and Yulia Tsvetkov. 2021. Selfexplain: A self-explaining architecture for neural text classifiers. *ArXiv*, abs/2103.12279.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. *Text mining: applications and theory*, 1:1–20.

François Rousseau and Michalis Vazirgiannis. 2015. Main core retention on graph-of-words for single-document keyword extraction. In *European Conference on Information Retrieval*, pages 382–393. Springer.

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.

Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. 2017. Learning important features through propagating activation differences. In *International Conference on Machine Learning*, pages 3145–3153. PMLR.

Takashi Tomokiyo and Matthew Hurst. 2003. A language model approach to keyphrase extraction. In *Proceedings of the ACL 2003 Workshop on Multiword Expressions: Analysis, Acquisition and Treatment - Volume 18*, MWE '03, page 33–40, USA. Association for Computational Linguistics.

Xiaojun Wan and Jianguo Xiao. 2008. Single document keyphrase extraction using neighborhood knowledge. In *AAAI*, volume 8, pages 855–860.

Wenbo Wang, Yang Gao, He-Yan Huang, and Yuxiang Zhou. 2019. Concept pointer network for abstractive summarization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3076–3085.

Carl Yang, Jieyu Zhang, Haonan Wang, Bangzheng Li, and Jiawei Han. 2020. Neural concept map generation for effective document classification with interpretable structured summarization. In *SIGIR*.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Yang Yu and Vincent Ng. 2018. Wikirank: Improving keyphrase extraction based on background knowledge. *arXiv preprint arXiv:1803.09000*.

8 Appendix

8.1 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. 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 λ parameter used to combine the phrase and context representations was fixed at 0.5. We tried values between 0.1 and 0.9 and did not find signifcant difference.

Dataset	Type	Split	Total docs	Avg words per doc	Avg keyphrases per doc
SciERC	Scientific	Train	350	130	16
		Dev	50	130	16
		Test	100	134	17
SciREX	Scientific	Train	306	5601	353
		Dev	66	5484	354
		Test	66	6231	387
SemEval17	Scientific	Train	350	160	21
		Dev	50	193	27
		Test	100	186	23
500N-KPCrowd	News	Train	400	430	193
		Dev	50	465	86
		Test	50	420	116
BBC News	News	All	2225	385	-
ICLR	Scientific	All	8317	6505	-

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;shortterm;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;paper 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;l;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.