

# UCPhrase: Unsupervised Context-aware Quality Phrase Tagging

Xiaotao Gu<sup>1\*</sup>, Zihan Wang<sup>2\*</sup>, Zhenyu Bi<sup>2</sup>, Yu Meng<sup>1</sup>, Liyuan Liu<sup>1</sup>, Jiawei Han<sup>1</sup>, Jingbo Shang<sup>2</sup>

<sup>1</sup>University of Illinois at Urbana-Champaign {xiaotao2, yumeng5, ll2, hanj}@illinois.edu

<sup>2</sup>University of California San Diego {ziw224, z1bi, jshang}@ucsd.edu

## ABSTRACT

Identifying and understanding *quality phrases* from context is a fundamental task in text mining. The most challenging part of this task arguably lies in uncommon, emerging, and domain-specific phrases. The infrequent nature of these phrases significantly hurts the performance of phrase mining methods that rely on sufficient phrase occurrences in the input corpus. Context-aware tagging models, though not restricted by frequency, heavily rely on domain experts for either massive sentence-level gold labels or handcrafted gazetteers. In this work, we propose UCPhrase, a novel unsupervised context-aware quality phrase tagger. Specifically, we induce high-quality phrase spans as silver labels from consistently co-occurring word sequences within each document. Compared with typical context-agnostic distant supervision based on existing knowledge bases (KBs), our silver labels root deeply in the input domain and context, thus having unique advantages in preserving contextual completeness and capturing emerging, out-of-KB phrases. Training a conventional neural tagger based on silver labels usually faces the risk of overfitting phrase surface names. Alternatively, we observe that the contextualized attention maps generated from a Transformer-based neural language model effectively reveal the connections between words in a surface-agnostic way. Therefore, we pair such attention maps with the silver labels to train a lightweight span prediction model, which can be applied to new input to recognize (unseen) quality phrases regardless of their surface names or frequency. Thorough experiments on various tasks and datasets, including corpus-level phrase ranking, document-level keyphrase extraction, and sentence-level phrase tagging, demonstrate the superiority of our design over state-of-the-art pre-trained, unsupervised, and distantly supervised methods.

## CCS CONCEPTS

• Computing methodologies → Information extraction.

## KEYWORDS

phrase mining; language models; unsupervised method

### ACM Reference Format:

Xiaotao Gu<sup>1\*</sup>, Zihan Wang<sup>2\*</sup>, Zhenyu Bi<sup>2</sup>, Yu Meng<sup>1</sup>, Liyuan Liu<sup>1</sup>, Jiawei Han<sup>1</sup>, Jingbo Shang<sup>2</sup>. 2021. UCPhrase: Unsupervised Context-aware Quality

\*Equal Contribution.



This work is licensed under a Creative Commons Attribution International 4.0 License.

KDD '21, August 14–18, 2021, Virtual Event, Singapore.

© 2021 Association for Computing Machinery.

ACM ISBN 978-1-4503-8332-5/21/08...\$15.00

<https://doi.org/10.1145/3447548.3467397>

Phrase Tagging. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '21), August 14–18, 2021, Virtual Event, Singapore*. ACM, New York, NY, USA, 9 pages. <https://doi.org/10.1145/3447548.3467397>

## 1 INTRODUCTION

Quality phrases refer to informative multi-word sequences that “*appear consecutively in the text, forming a complete semantic unit in certain contexts or the given document*” [10]. Identifying and understanding quality phrases from context is a fundamental task in text mining. Automated quality phrase tagging serves as a cornerstone in a broad spectrum of downstream applications, including but not limited to entity recognition [35], text classification [1], and information retrieval [5].

The most challenging open problem in this task is how to recognize uncommon, emerging phrases, especially in specific domains. These phrases are essential in the sense of their significant semantic meanings and the large volume—following a typical Zipfian distribution, uncommon phrases can add up to a significant portion of quality phrases [37]. Moreover, emerging phrases are critical in understanding domain-specific documents, such as scientific papers, since new terminologies often come along with transformative innovations. However, mining such sparse long-tail phrases is nontrivial, since a frequency threshold has long ruled them out in traditional phrase mining methods [6, 8, 23, 25, 34] due to the lack of reliable frequency-related corpus-level signals (e.g., the mutual information of its sub-ngrams). For instance, AutoPhrase [34] only recognizes phrases with at least 10 occurrences by default.

For infrequent phrases, the tagging process largely relies on local context. Recent advances in neural language models have unleashed the power of sentence-level contextualized features in building chunking- and tagging-based models [27, 36]. These context-aware models can even recognize unseen phrases from new input texts, thus being no longer restricted by frequency. However, training a domain-specific tagger of reasonably high quality requires expensive, hard-to-scale effort from domain experts for massive sentence-level gold labels or handcrafted gazetteers.

In this work, we propose UCPhrase, a novel unsupervised context-aware quality phrase tagger. It first induces high-quality silver labels directly from the corpus under the unsupervised setting, and then trains a tailored Transformer-based neural model that can recognize quality phrases in new sentences. Figure 1 presents an overview of UCPhrase. The two major steps are detailed as follows.

By imitating the reading process of humans, we derive supervision directly from the input corpus. Given a document, human readers can quickly recognize new phrases or terminologies from the consistently used word sequences *within the document*. The “document” here refers to a collection of sentences centered on the same topic, such as sentences from an abstract of a scientific paper

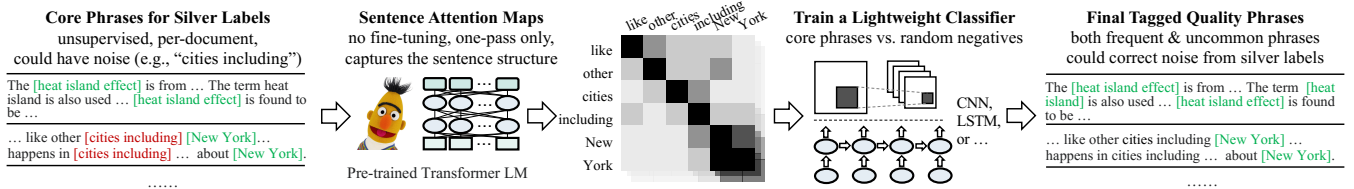


Figure 1: An overview of our UCPhrase: unsupervised context-aware quality phrase tagging.

and tweets mentioning the same hashtag. Inspired by this observation, we propose to extract **core phrases** from each document, which are maximal contiguous word sequences that appear in the document more than once. The “maximal” here means that if one expands this word sequence further towards the left or right, its frequency within this document will drop. To avoid uninformative phrases (e.g., “of a”), we conduct simple filtering of stopwords before finalizing the silver labels. Note that our proposed silver label generation follows a per-document manner. Therefore, compared with typical context-agnostic distant supervision based on existing knowledge bases or dictionaries [34–36], our supervision roots deeply in the input domain and context, thus having unique advantages in preserving contextual completeness of matched spans and capturing much more emerging phrases.

We further design a tailored neural tagger to fit our silver labels better. Training a conventional neural tagger based on silver labels usually faces a high risk of overfitting the observed labels [24]. With access to the word-identifiable embedding features, it is easy for the model to achieve nearly zero training error by rigidly memorizing the surface names of training labels. Alternatively, we find that the contextualized attention distributions generated from a Transformer-based neural language model could capture the connections between words in a surface-agnostic way [19]. Intuitively, the attention maps of quality phrases should reveal distinct patterns from ordinary word spans. Moreover, attention-based features block the direct access to the surface names of training labels, and force the model to learn about more general context patterns. Therefore, we pair such surface-agnostic features based on attention maps with the silver labels to train a neural tagging model, which can be applied to new input to recognize (unseen) quality phrases. Specifically, given an unlabeled sentence of  $N$  words, we first encode the sentence with a pre-trained Transformer-based language model and obtain the attention maps as features. The  $N \times N$  matrices from different Transformer layers and attention heads can be viewed as images to be classified with multiple channels. A lightweight CNN-based classifier is then trained to distinguish quality phrases from randomly sampled negative spans.

Thorough experiments on various tasks and datasets, including corpus-level phrase ranking, document-level keyphrase extraction, and sentence-level phrase tagging, demonstrate the superiority of our design over state-of-the-art unsupervised, distantly supervised methods, and pre-trained off-the-shelf tagging models. It is noteworthy that our trained model is robust to the noise in the core phrases—our case studies in Section 4.7 show that the model can identify inferior training labels by assigning extremely low scores.

Efficiency wise, thanks to the rich semantic and syntactic knowledge in the pre-trained language model, we can simply use the generated attention maps as informative features without fine-tuning

the language model. Hence we only need to update the lightweight classification model during training, making the training process as fast as one inference pass of the language model through the corpus with limited resource consumption.

To the best of our knowledge, UCPhrase is the first unsupervised context-aware quality phrase tagger. It enjoys the rich knowledge from the pre-trained neural language models. The learned phrase tagger works efficiently and effectively without reliance on human annotations, existing knowledge bases, or phrase dictionaries. We summarize our key contributions as follows:

- We propose to mine silver labels that root deeply in the input domain and context by recognizing core phrases, i.e., maximal word sequences that occur consistently in a per-document manner.
- We propose to replace the conventional contextualized word representations with surface-agnostic attention maps generated by pre-trained Transformer-based language models to alleviate the risk of overfitting silver labels.
- We conduct extensive experiments, ablation studies, and case studies to compare UCPhrase with state-of-the-art unsupervised, distantly supervised methods, and pre-trained off-the-shelf tagging models. The results verify the superiority of our method <sup>1</sup>.

## 2 PROBLEM DEFINITION

Given a sequence of words  $[w_1, \dots, w_N]$ , a quality phrase is a contiguous span of words  $[w_i, \dots, w_{i+k}]$  that form a complete and informative semantic unit in context. Though some studies also view unigrams as potential phrases, in this work, we focus on multi-word phrases ( $k > 0$ ), which are more informative, yet more challenging to get due to both diversity and sparsity.

To effectively capture phrases with potential overlaps, e.g., “*information extraction*” in “*information extraction systems*”, we adopt the span prediction framework, where each possible span in the sentence is assigned a binary label. To avoid a quadratic growth of the size of candidate spans, we follow previous work [25, 34] to set a maximum span length  $K$ . We also explore alternative classifiers based on the sequence labeling framework in Section 3.

## 3 UCPhrase: METHODOLOGY

Figure 1 presents an overview of UCPhrase. As an unsupervised method, UCPhrase first mines core phrases directly from each document as silver labels and extracts surface-agnostic attention features with a pre-trained language model. A lightweight classifier is then trained with silver labels and randomly sampled negative labels. Algorithm 1 shows the detailed training process.

<sup>1</sup>Code and data: <https://github.com/xgeric/UCPhrase-exp>.

### Distant Supervision based on Wiki Entities

**Doc1:** ... study about heat [island effect] ... The heat [island effect] arises because the buildings...of their heat [island effect]...

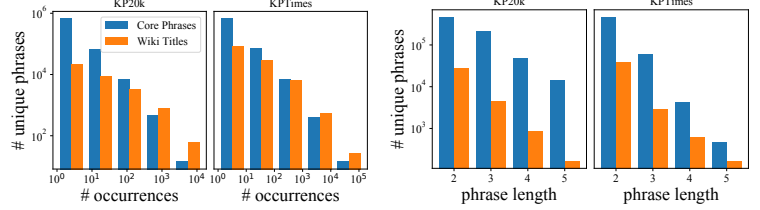
**Doc2:** ... propose to extract core phrases ... robust to potential noise in core phrases ... the surface names of core phrases...

### Core Phrase Mining

**Doc1:** ...a study about [heat island effect]... The [heat island effect] arises because the buildings...of their [heat island effect]...

**Doc2:** ...propose to extract [core phrases]... robust to potential noise in [core phrases]... the surface names of [core phrases]...

(a) Example Silver Labels



(b) Occurrence Distribution

(c) Length Distribution

**Figure 2: Comparing core phrases with context-agnostic distant supervision. (a) An illustrative example with context. Our core phrases preserve better contextual completeness and discover emerging new concepts introduced in the document. (b) Distributions of the generated silver labels with their occurrences in the corpus. The X-axis represents bins of phrase occurrences in the corpus. The Y-axis (exponential) represents the number of unique phrases in each bin. (c) Distributions of the generated silver labels with their lengths (# of words).**

### Algorithm 1: UCPhrase: unsupervised model training

---

**Input:** A corpus of  $M$  unlabeled documents,  $\{d_m\}_{m=1}^M$ ; a pre-trained language model (LM); the attention-based lightweight classification model  $f(\cdot; \theta)$  to be trained.

// Generate silver labels as supervision (Sec. 3.1)

**for each document  $d_m$  do**

- $\mathcal{P}_m^+ = \text{MineCorePhrases}(d_m)$ .
- Randomly sample remaining spans as negative set  $\mathcal{P}_m^-$ .

// Generate surface-agnostic features using LM (Sec. 3.2)

**for each labeled span  $p_i \in \mathcal{P} = \bigcup_{m=1}^M \{\mathcal{P}_m^-, \mathcal{P}_m^+\}$  do**

- $X_{p_i} = \text{ExtractAttentionFeature}(\text{LM}, p_i)$

$\mathcal{D}_{\text{train}}, \mathcal{D}_{\text{valid}} = \text{split}(X_{\mathcal{P}}, \mathcal{P})$ .

// Train a classifier based on features & labels (Sec. 3.3)

**repeat**

- Sample a minibatch  $\{X_{\mathcal{P}_b}, \mathcal{P}_b\}$  from  $\mathcal{D}_{\text{train}}$ .
- Update model parameters with loss  $\ell(f(X_{\mathcal{P}_b}; \theta), \mathcal{P}_b)$ .

**until  $F_1$  score on  $\mathcal{D}_{\text{valid}}$  drops;**

**Output:** The trained classification model  $f$ .

---

## 3.1 Silver Label Generation

As the first step, we seek to collect high-quality phrases in the input corpus following an unsupervised way, which will be our silver labels for the tagging model training. A common practice for automated label fetching is to conduct a context-agnostic matching between the corpus and a given quality phrase list, either mined with unsupervised models or collected from an existing knowledge base (KB). Such methods, as we show later, can suffer from incomplete labels due to the negligibility of context.

On the contrary, based on the definition of phrases, we look for consistently used word sequences in context. We propose to treat documents as context and collect high-quality phrases directly from each document. The “document” here refers to a collection of sentences centered on the same topic, such as sentences from an abstract of a scientific paper and tweets mentioning the same hashtag. This way, we expect to preserve better contextual completeness that reflects the original writing intention.

We view a document  $d_m$  as a contiguous word sequence  $[w_1, \dots, w_N]$  and then mine max contiguous sequential patterns. A valid pattern here is a word span  $[w_i, \dots, w_j]$  that appear more than once in the input sequence. One can easily adjust the frequency threshold to balance the quality and quantity of valid patterns.

In this work, we simply use the minimum requirement of two occurrences without further tuning, and find it works well for both short documents like paper abstracts and long documents like news reports. To preserve completeness, we only leave max patterns that are not sub-patterns of any other valid patterns. Uninformative patterns like “of a” are removed with a stopwords list widely used by previous work [25, 34]. We treat the remaining max patterns as core phrases of document  $d_m$ , and add them to the positive training samples  $\mathcal{P}_m^+$ . An equal number of negative samples are randomly drawn from the remaining spans in  $d_m$ , denoted as  $\mathcal{P}_m^-$ .

Figure 2 compares the silver labels generated by our core phrases with those by distant supervision, which follows a context-agnostic string matching from the Wikipedia entities. From the real example in Figure 2(a), “heat island effect” is not a Wikipedia entity, but “island effect” is one. Distant supervision hence generates a flawed label by favoring the popular, incomplete phrases and partially matching the real phrase. Similar examples are quite common especially when it comes to compound phrases, like “biomedical data mining”. On the contrary, our core phrase mining can generate labels with better contextual completeness. Core phrase mining can also dynamically capture concepts or expressions newly introduced in each document, such as the “core phrases” in this paper. Figure 2(b) confirms this by showing the distribution of unique phrases of the two types of silver labels, mined from the KP20k CS publication corpus and the KPTime news corpus, with respect to their frequency. In particular, core phrase mining discovers more unique phrases with less than 10 occurrences in the corpus (30x on KP20k, 9x on KPTime). As Figure 2(c) demonstrates, core phrases outnumber matched Wiki titles on all length ranges. Overall, core phrase mining discovers much more unique phrases than distant supervision (20x on KP20k, 6x on KPTime).

Of course, there also inevitably exist noises in mined core phrases due to random word combinations consistently used in some documents, e.g., “countries including”. Fortunately, since we collected core phrases from each document independently, such noisy labels will not spread and be amplified to the entire corpus. In fact, among the tagged core phrases randomly sampled from two datasets, the overall proportion of high-quality labels is over 90%. The large volume of reasonably high-quality silver labels provides a robust foundation for us to train a span classifier that learns about general context patterns to distinguish noisy spans. As Section 4.7 shows, the final classifier can assign extremely low scores to false-positive

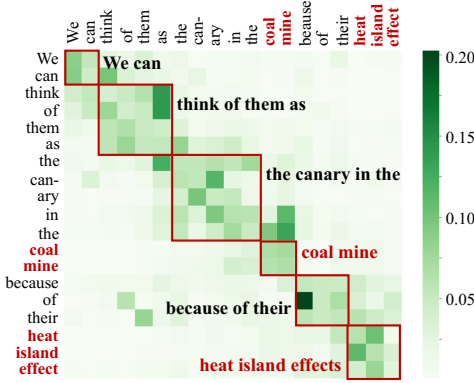


Figure 3: Illustration of the attention map generated by one of the pre-trained RoBERTa layers, averaged over all attention heads.

phrases in training labels.

In summary, document-level core phrase mining provides a simple and effective way to automatically fetch abundant context-aware silver labels of reasonably good quality without relying on external KBs. In ablation studies (Section 4.6) we show that models trained with such free silver labels can outperform the same models trained with distant supervision.

### 3.2 Surface-agnostic Feature Extraction

To build an effective context-aware tagger for quality phrases, in addition to labels, we need to figure out the contextualized feature representation for each span. Traditional word-identifiable features (e.g., contextualized embedding) make it easy for a classification model to overfit the silver labels by rigidly memorizing the surface names of training labels. A degenerated name-matching model can easily achieve zero training error without really learning about any useful, generalizable features.

In principle, the recognition of a phrase should depend on the role that it plays in the sentence. Kim et al. [19] show that the structure information of a sentence can be largely captured by its attention distribution. Therefore, we propose to obtain surface-agnostic features from the attention distributions generated by a pre-trained Transformer-based language model encoder (LM), such as BERT [7] and RoBERTa [26].

Given a sentence  $[w_1, \dots, w_N]$ , we encode it with a language model pre-trained on a corpus from the same domain (e.g., scientific domain). Suppose this language model has  $L$  layers, and each layer has  $H$  attention heads. Each attention head  $h$  from layer  $l$  produces an attention map  $A^{l,h} \in \mathbb{R}^{N \times N}$  of the sentence. We denote the aggregated attention map from attention heads of all layers as  $A \in \mathbb{R}^{N \times N \times (H \cdot L)}$ , where  $A_{i,j} \in \mathbb{R}^{H \cdot L}$  is a vector that contains the attention scores from  $w_i$  to  $w_j$ . Finally, for each candidate span  $p = [w_i, \dots, w_j]$ , we denote its feature as  $X_p = A_{i \dots j, i \dots j}$ .

Ideally, the attention maps of quality phrases should reveal distinct patterns of word connections. Figure 3 shows a real example of the generated attention map of a sentence. The chunks on the attention map lead to a clear separation of different parts of the sentence. From all chunks, our final span classifier (Section 3.3) accurately distinguishes the quality phrases (“coal mine”, “heat island effects”) from ordinary spans (e.g., “We can”), indicating the informativeness of the attention features.

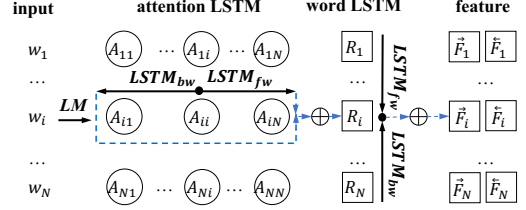


Figure 4: An alternative classifier based on attention-level LSTM.

**Efficient Implementation.** Thanks to the rich syntactic and semantic knowledge in the pre-trained language model, the generated attention maps are already informative enough for phrase tagging. In this work, we adopt the RoBERTa model [26], one of the state-of-the-art Transformer-based language models, as a feature extractor without the need for further fine-tuning. We only need to apply the pre-trained RoBERTa model for one inference pass through the target corpus for feature extraction.

The overall efficiency now mainly depends on the size of the attention map, which is  $N \times N \times (H \cdot L)$ .  $N$  is restricted to the length of each span during training, and for inference, we apply sentence-level encoding, with each sentence restricted to at most 64 tokens. Depth wise, existing studies have observed considerable redundancy in the outputs of different Transformer layers, including attention distributions [14, 15]. For this reason, as the default setting of UCPhrase, we only preserve attention maps from the first 3 layers in RoBERTa (i.e.,  $L = 3$ ). As RoBERTa has 12 layers in total, this saves 75% of resource consumption. We have quantitatively compared the final tagging performance of using 3 layers vs. using all 12 layers in Section 4.6. As the experimental results suggest, using 3 layers exhibits comparable performance with the full model.

### 3.3 Lightweight Span Classifier

With the labels and features in-house, we are ready to build a classifier to recognize spans of quality phrases. Our framework is general and compatible with various classification models. For the sake of efficiency, we wish to find a lightweight classifier.

Given the attention map of a  $k$ -word span, an accurate classifier should effectively capture inter-word relationships from different LM layers and at different ranges. Naturally, the attention map can be viewed as a square image of  $k$  pixels for both height and width, with  $H \cdot L$  channels. We can now transform the phrase classification problem into an image classification problem: given a multi-channel image (attention map), we want to predict whether the corresponding word span is a quality phrase. Specifically, we apply a two-layer convolutional neural network (CNN) model on the multi-channel attention map. The output is then fed to a logistic regression layer to assign a binary label for the corresponding span. During the training process, the classification model  $f(\cdot; \theta)$  parameterized by  $\theta$  is learned by minimizing the loss over the training set  $\{X_P, \mathcal{P}\}$ :

$$\hat{\theta} = \underset{\theta}{\operatorname{argmin}} \frac{1}{|\mathcal{P}|} \sum_{i=1}^{|\mathcal{P}|} \ell(p_i, f(X_{p_i}; \theta)),$$

where  $p_i \in \mathcal{P} = \bigcup_{m=1}^M \{\mathcal{P}_m^-, \mathcal{P}_m^+\}$  represents the  $i$ -th labeled span, and  $\ell$  is the binary cross entropy loss function. The model is updated with minibatch-based stochastic gradient descent.

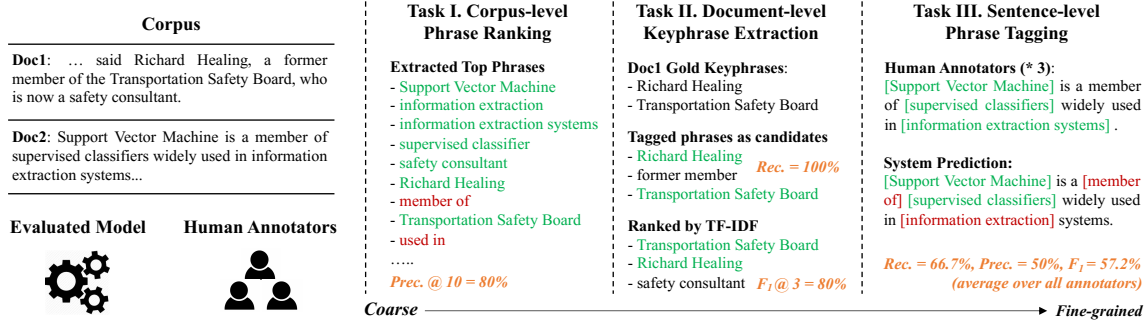


Figure 5: Illustration of the Three Evaluation Tasks and their Evaluation Metrics.

Note that  $\theta$  here only includes the parameters in the two CNN layers and the logistic regression layer during training, which makes the training process efficient in terms of resource consumption. In fact, the checkpoint of training parameters from each epoch can be stored in a 22 KB file on disk.

**Alternative Classifiers.** In our study, we also considered some alternative classifiers. One intuitive choice here is LSTM-based models following the sequence labeling framework. We illustrate the general idea in Figure 4. Specifically, for sentence  $[w_1, w_2, \dots, w_N]$ , we first encode the attention map  $\mathbf{A} \in \mathbb{R}^{N \times N \times (H \cdot L)}$  with forward and backward runs of LSTM to get an attention-based word representation through the final output of both LSTMs as follows,

$$\begin{aligned}\vec{\mathbf{R}}_i &= \text{LSTM}(\mathbf{A}_{i,i}, \mathbf{A}_{i,i+1}, \dots, \mathbf{A}_{i,N})_{\text{last}}, \\ \overleftarrow{\mathbf{R}}_i &= \text{LSTM}(\mathbf{A}_{i,i}, \mathbf{A}_{i,i-1}, \dots, \mathbf{A}_{i,1})_{\text{last}}, \\ \mathbf{R}_i &= [\vec{\mathbf{R}}_i, \overleftarrow{\mathbf{R}}_i], \quad 1 \leq i \leq N.\end{aligned}$$

Another bidirectional LSTM layer is built upon the word representations  $\mathbf{R}$  to extract the feature  $\mathbf{F}$ , i.e.,

$$\begin{aligned}\vec{\mathbf{F}}_{1,2,\dots,N} &= \text{LSTM}(\mathbf{R}_1, \mathbf{R}_2, \dots, \mathbf{R}_N), \\ \overleftarrow{\mathbf{F}}_{N,N-1,\dots,1} &= \text{LSTM}(\mathbf{R}_N, \mathbf{R}_{N-1}, \dots, \mathbf{R}_1).\end{aligned}$$

Scheme wise, there are two popular labeling schemes in sequence labeling: (1) *Tie-or-Break*, which is predicting whether each consecutive pair of words belong to the same phrase, and (2) *BIO Tagging*, which is tagging phrases in the sentence through a Begin-Inside-Outside scheme [32]. We are not using BIOES [33] as we focus on multi-word phrases. For the *Tie-or-Break tagging scheme*, we apply a 2-layer Multi-layer Perceptron followed by a Sigmoid classification function to predict whether the  $[\vec{\mathbf{F}}_i, \overleftarrow{\mathbf{F}}_{i+1}]$  representation corresponds to a tie between word  $i$  and word  $i+1$  or a break. For the *BIO tagging scheme*, in a word-wise manner, we concatenate the representations  $\vec{\mathbf{F}}_{1,2,\dots,N}$  and  $\overleftarrow{\mathbf{F}}_{N,N-1,\dots,1}$  into  $N$  representations for each sentence, and then send them through a Conditional Random Field (CRF) layer [18, 22] to predict the BIO tags for phrases.

Other training procedures for both of the classifiers are the same as the aforementioned default span classifier. These alternative classifiers have comparable performance, as confirmed in Section 4.6.

## 4 EXPERIMENTS

We compare our UCPhrase with previous studies on multi-word phrase mining tasks on two datasets and three tasks at different

Table 1: Dataset statistics on KP20k and KPTimes.

Statistics	KP20k	KPTimes
	<i>Train Set</i>	
# documents	527,090	259,923
# words per document	176	907
	<i>Test Set</i>	
# documents	20,000	20,000
# multi-word keyphrases	37,289	24,920
# unique	24,626	8,970
# absent in training corpus	4,171	2,940

granularity: corpus-level phrase ranking, document-level keyphrase extraction, and sentence-level phrase tagging.

### 4.1 Evaluation Tasks and Metrics

We evaluate all methods on the following three tasks. Figure 5 illustrates the tasks and evaluation metrics with some examples.

**Task I: Phrase Ranking** is a popular evaluation task in previous statistics-based phrase mining work [6, 8, 23, 25, 34]. Specifically, it evaluates the “global” rank list of phrases that a method finds from the input corpus. Since UCPhrase does not explicitly compute a “global” score for each phrase, we use the average logits of all occurrences of a predicted phrase to rank phrases.

In our experiments, for each method<sup>2</sup> on each dataset, we quantitatively evaluate the precision of the phrases found in the top-ranked 5,000 and 50,000 phrases, denoted as **P@5K** and **P@50K**. Since it is expensive to hire annotators to annotate all these phrases, we estimate the precision scores by randomly sampling 200 phrases from rank lists. Extracted phrases from different methods are shuffled and mixed before presenting to the annotators.

**Task II: Keyphrase Extraction** is a classic task to extract salient phrases that best summarize a document [9], which essentially has two stages: candidate generation and keyphrase ranking. At the first stage, we treat all compared methods as candidate phrase extractors and evaluate the **recall** of generated candidates. In each document, the recall measures how many gold keyphrases are extracted in the candidate list. For fair comparison, we preserve the same number of candidates from the rank list of each method for evaluation.

For the end-the-end performance, we apply the classic TF-IDF model to rank the candidate phrases extracted by different methods. In each document, we follow the standard evaluation method [13] to calculate the  $F_1$  score of the top-10 ranked phrases (**F<sub>1</sub>@10**). The reported recall and  $F_1$  scores are averaged in a macro way across

<sup>2</sup>Except for methods that does not report any form of scores for ranking.

Table 2: Evaluation results (%) of three tasks for all compared methods on datasets on two domains.

Method Type	Method Name	Task I: Phrase Ranking				Task II: KP Extract.				Task III: Phrase Tagging					
		KP20k		KPTimes		KP20K		KPTimes		KP20k		KPTimes			
		P@5K	P@50K	P@5K	P@50K	Rec.	F <sub>1</sub> @10	Rec.	F <sub>1</sub> @10	Prec.	Rec.	F <sub>1</sub>	Prec.	Rec.	F <sub>1</sub>
Pre-trained	PKE [3]	–	–	–	–	57.1	12.6	61.9	4.4	54.1	63.9	58.6	56.1	62.2	59.0
	Spacy [17]	–	–	–	–	59.5	15.3	60.8	8.6	56.3	68.7	61.9	61.9	62.9	62.4
	StanfordNLP [27]	–	–	–	–	51.7	13.9	60.8	8.7	48.3	60.7	53.8	56.9	60.3	58.6
Distantly Supervised	AutoPhrase [34]	97.5	96.0	96.5	95.5	62.9	18.2	77.8	10.3	55.2	45.2	49.7	44.2	47.7	45.9
	Wiki+RoBERTa	<b>100.0</b>	<b>98.5</b>	<b>99.0</b>	<b>96.5</b>	<b>73.0</b>	19.2	64.5	9.4	58.1	64.2	61.0	60.9	65.6	63.2
Unsupervised	TopMine [8]	81.5	78.0	85.5	71.0	53.3	15.0	63.4	8.5	39.8	41.4	40.6	32.0	36.3	34.0
	UCPhrase (ours)	96.5	96.5	96.5	95.5	72.9	<b>19.7</b>	<b>83.4</b>	<b>10.9</b>	<b>69.9</b>	<b>78.3</b>	<b>73.9</b>	<b>69.1</b>	<b>78.9</b>	<b>73.5</b>

all documents in the same dataset.

**Task III: Phrase Tagging** is a fine-grained task that aims to find all occurrences of phrases in sentences. Specifically, it evaluates the extracted phrase spans for each sentence. We treat each phrase mining method as a sentence tagger that identifies starting and ending boundaries of phrases in a sentence. We randomly sample 200 sentences on each dataset and ask three annotators to tag all spans of multi-word phrases. Each sentence is annotated by all annotators independently, and the agreement between human annotations is around 90%. We then pool all annotations together, evaluate the predicted spans, and report the overall **precision**, **recall** and **F<sub>1</sub>** scores. Note that these scores are computed in a micro average fashion following previous work on entity recognition [35].

## 4.2 Datasets

We adopt two commonly used datasets from different domains to evaluate all different methods.

- **KP20k** [29] is a collection of *titles & abstracts* from Computer Science papers—527,090 for training and 20,000 for testing.
- **KPTimes** [12] consists of *news articles* on New York Times from 2006 to 2017, supplemented with 10,000 more news articles from Japan Times. In total, there are 259,923 articles for training, and 20,000 articles for testing.

Following Gururangan et al. [16], sentence separation and tokenization is conducted with Spacy [17] for postprocessing. All three tasks are evaluated on the test sets. Statistics of the two datasets are shown in Table 1. Note that 17% unique keyphrases in the test of KP20k never occur in the corresponding training corpus. On KPTimes the absence ratio is 33%. Hence, the task can be challenging for models relying on phrase frequencies and models rigidly memorizing training phrases.

## 4.3 Compared Methods

We compare the proposed method with existing methods under the same scenario, where no gold annotations for training are available. This leads to three categories: unsupervised phrase mining methods, distantly supervised methods with an existing KB, and pre-trained off-the-shelf toolkits. For each method that requires training (i.e., all the unsupervised and distantly supervised ones), we use the unlabeled documents from the training set for model learning.

For *unsupervised methods* we consider:

- **ToPMine** [8], the state-of-the-art unsupervised phrase mining method building upon statistical features.
- **UCPhrase**, the proposed method in this work.

For *distantly supervised methods*, we use silver labels generated from the *Wiki Entities*, which is firstly used in [34].

- **AutoPhrase** [34] leverages statistics-based phrase classifier and further enhances it with a POS-guided phrasal segmentation model for sentence tagging and phrase frequency rectification.
- **Wiki+RoBERTa** is a strong baseline that we propose here. It can be viewed as a variant of UCPhrase with the same span prediction framework and the same pre-trained LM as our method but following distant supervision. Also, it uses the output states from the last layer of the pre-trained RoBERTa as feature instead of attention maps. As shown in [24], stopping the model training early is an essential intervention for distantly supervised tagging models. To fully unleash the potential of the Wiki+RoBERTa baseline, we *manually stop* its training process *after the first epoch* to avoid overfitting. This indeed achieves a better test performance than stopping after more epochs.

For *off-the-shelf toolkits* we consider the linguistic-based methods that are pre-trained with labeled pos-tagging or parsing data.

- **PKE** [3] is a widely used toolkit for keyphrase extraction. Its phrase mining module is a chunking model based on a supervised POS-tagging model from NLTK [2] and a set of grammar rules.
- **Spacy** [17] is an industrial library with a pre-trained phrase chunking model based on supervised POS tagging and parsing.
- **StanfordCoreNLP** [27] is a long recognized NLP package whose chunking model is based on dependency parsing.

## 4.4 Reproduction Details

For KPTimes, we use the official RoBERTa model pre-trained on documents from the general domain. On the KP20k dataset, we use the “allenai/cs\_roberta\_base” RoBERTa model [16]<sup>3</sup>. The model is based on the standard pre-trained RoBERTa model, and then trained on unlabeled Computer Science publications. This domain-adapted model performs slightly better on the KP20k dataset than the original model. We adopt the Adam [20] optimizer with the default parameters for model training. The learning rate is set to 0.001. As described in Algorithm 1, we train the classifier until its performance on the 10% hold-out validation set  $\mathcal{D}_{valid}$  drops. Other details have been covered in Section 3. We will publish our data and code base for reproduction.

## 4.5 Evaluation Results

From Table 2 we can see that UCPhrase achieves the best overall performance on all the three evaluation tasks. The performance

<sup>3</sup> [https://huggingface.co/allenai/cs\\_roberta\\_base](https://huggingface.co/allenai/cs_roberta_base)

Table 3: Ablation study of UCPhrase model variants (%).

	Design Choices			KP Extract.				Phrase Tagging					
				KP20k		KPTimes		KP20k			KPTimes		
	supervision	feature	fine-tune	Rec.	F <sub>1</sub> @10	Rec.	F <sub>1</sub> @10	Prec.	Rec.	F <sub>1</sub>	Prec.	Rec.	F <sub>1</sub>
UCPhrase	core	attention	no	72.9	<b>19.7</b>	<b>83.4</b>	<b>10.9</b>	69.9	<b>78.3</b>	<b>73.9</b>	<b>69.1</b>	<b>78.9</b>	<b>73.5</b>
Variants	Wiki	attention	no	68.7	17.7	79.4	10.7	<b>72.1</b>	71.9	72.0	64.1	67.6	65.8
	Wiki	embedding	no	73.0	19.2	64.5	9.4	60.9	65.6	63.2	60.9	65.6	63.2
	core	embedding	no	79.3	<b>19.7</b>	78.7	10.2	68.4	74.6	71.4	55.7	64.8	59.9
	core	embedding	yes	<b>80.3</b>	<b>19.7</b>	73.9	9.9	68.6	74.8	71.6	53.3	64.5	59.0

Table 4: Comparison of attention feature aggregated from different numbers of Transformer layers, evaluated on KP20k (%).

# Layers	KP Extract.		Phrase Tagging		
	Rec.	F <sub>1</sub> @10	Prec.	Rec.	F <sub>1</sub>
3	72.9	19.7	<b>69.9</b>	<b>78.3</b>	<b>73.9</b>
12	<b>81.8</b>	<b>20.6</b>	69.4	76.8	72.9

gap becomes more vivid as the task becomes more fine-grained.

In the corpus-level phrase ranking task, most methods show very high precision (i.e.,  $\geq 95\%$ ) on the top 50,000 mined phrases from each dataset. Notably, UCPhrase significantly outperforms the only other unsupervised method ToPMine and is able to perform on par with distantly supervised methods.

In the document-level keyphrase extraction task, UCPhrase has better recall than most compared methods, demonstrating a coverage of high-quality phrases. Wiki+RoBERTa has slightly better recall on the KP20k dataset (0.1%) within a reasonable range, considering Wiki+RoBERTa has access to hundreds of thousands keyphrases from Wiki Entities. Note that UCPhrase outperforms all the compared methods on the end-to-end performance (i.e.,  $F_1@10$ ), which verifies its value to the application of keyphrase extraction.

In the sentence-level phrase tagging task, UCPhrase achieves  $F_1$  scores of more than 73% on both datasets, showing significant advantages (i.e.,  $> 10\%$  in  $F_1$ ) over all the compared methods. This is truly encouraging given the facts that (1) UCPhrase is an unsupervised phrase mining model that requires no human effort, and (2) even human annotators cannot fully agree with each other on some particular phrases, and have around 10% disagreement on this task. This phrase tagging task makes clear that UCPhrase is able to find phrases much more accurately than compared methods. In Section 4.6 we apply comprehensive comparison between different models on real examples, for a more straightforward visualization of the pros and cons of compared methods.

#### 4.6 Ablation studies

To gain deeper insights, we apply extensive ablation studies to test model variants from several aspects, as summarized in Table 3. For supervision, we compare the silver labels generated by unsupervised core phrase mining (*core*), and those generated by distant supervision with Wikipedia entities (*Wiki*). For the type of features, we compare the attention map features (*attention*), and the output states of RoBERTa (*embedding*).

**Supervision: Core Phrase vs. Distant Supervision.** When using the same type of feature, unsupervised models with core phrases as supervision significantly outperform distantly supervised models on most metrics by a clear gap. The better completeness and

Table 5: Exploring LSTM-based classifiers as alternatives based on Tie-or-Break and BIO labeling schemes, evaluated on KP20k (%).

Classifier	KP Extract.		Phrase Tagging		
	Rec.	F <sub>1</sub> @10	Prec.	Rec.	F <sub>1</sub>
CNN (default in UCPhrase)	68.1	18.7	69.9	<b>78.3</b>	<b>73.9</b>
LSTM w/ Tie-or-Break	<b>72.4</b>	<b>19.3</b>	68.1	72.3	70.1
LSTM w/ BIO	66.2	18.1	<b>71.0</b>	76.7	73.7

larger volume of core phrases bring unique advantages in training context-aware tagging models, not to mention the labels are fetched from the corpus for free without relying on an external KB. Moreover, the better diversity of core phrases effectively alleviates the risk of overfitting. It is also worth mentioning that different from the distantly supervised embedding model, the embedding-based model trained with core phrases does not require any manual early stopping to achieve satisfying performance.

**Features: Attention vs. Embedding.** When using the same type of supervision, models with attention features are almost always better than embedding-based features. This verifies our intuition that word-identifiable embeddings allow the classifier to easily overfit silver labels, while the surface-agnostic attention features force the model to learn about informative contextual features, and thus having a better ability of generalization.

**Attention: First Few Layers vs. Full Layers.** Table 4 compares UCPhrase trained with attention features aggregated from the first 3 layers of RoBERTa and those aggregated from all 12 layers, with intuitions explained in Section 3.2. The two models achieve comparable performance, while the small model only requires 25% resource consumption.

**Alternative Classifiers.** Table 5 compares our model with the the Tie-or-Break classifier and the BIO classifier as introduced in Section 3.3. Overall, the alternative classifiers have comparable performances, indicating the ability of our proposed method to generalize to different tagging schemes and model architectures.

#### 4.7 Case Studies

In spite of the reasonably high quality of the silver labels, we are curious about whether our final span classifier is robust to the noisy silver labels. To this end, we feed the silver labels to the span classifier and investigate the predicted probability scores. Table 7 presents the silver labels with probabilities below 1% and above 99% respectively. As it shows, our classifier successfully distinguishes high-quality core phrases from noisy spans, including typos (*italic font*) that happen to be used consistently in some document. The classifier draws a clear line between these two kinds of spans based

Table 6: Sentences tagged with different methods described in Section 4.3.

	KP20k	KPTimes
<b>Spacy</b>	We are interested in improving the Varshamov bound for [finite values] of length $n$ and [minimum distance] $d$ . We employ a [counting lemma] to this end which we find particularly useful in relation to [Varshamov graphs] .	The [United States] , at least theoretically , taxes companies on their [global profits] . But companies with a lot of [intellectual property] – notably [technology and pharmaceutical companies] – get away with paying a fraction of that amount .
<b>AutoPhrase</b>	We are interested in improving the [Varshamov bound] for finite values of length $n$ and [minimum distance] $d$ . We employ a [counting lemma] to this end which we find particularly useful in relation to Varshamov graphs .	The [United States] , at least theoretically , taxes companies on their global profits . But companies with a lot of [intellectual property] – notably [technology and pharmaceutical companies] – get away with paying a fraction of that amount .
<b>RoBERTa</b>	We are interested in improving the Varshamov bound for finite values of length $n$ and minimum distance $d$ . We employ a [counting lemma] to this end which we find particularly useful in relation to Varshamov graphs .	The [United States] , at least theoretically , [taxes companies] on their [global profits] . [But companies] with a lot of [intellectual property] – notably technology and [pharmaceutical companies] – get away with paying a fraction of that amount .
<b>UCPhrase</b>	We are interested in improving the [Varshamov bound] for [finite values] of length $n$ and [minimum distance] $d$ . We employ a [counting lemma] to this end which we find particularly useful in relation to [Varshamov graphs] .	The [United States] , at least theoretically , taxes companies on their [global profits] . But companies with a lot of [intellectual property] – notably technology and [pharmaceutical companies] – get away with paying a fraction of that amount .

on their attention features, which reflect their distinct roles in sentences. We have attempted to remove the low-score ones from the silver labels and re-train the classifier, however, the final performance changes little. This further verifies the robustness of our model, and its ability to capture general context features rather than rigid memorization.

Table 6 presents sentences tagged with representative methods from each category. As it shows, pre-trained models like Spacy can hardly adapt to a new domain without human annotations. For instance, it fails to recognize “*Varshamov bound*” as a phrase for recognizing “*bound*” as a verb. Statistics-based methods like AutoPhrase tend to miss uncommon phrases in the corpus, such as “*Varshamov graphs*”, “*finite values*”, and “*global profiles*”. The widely used distantly supervised methods based on word representations from a pre-trained language model (e.g., RoBERTa) can easily overfit the phrases in the KB, even though we have applied manual early stopping. The consequence of rigid memorization comes in two folds. First, the model can miss a lot of out-of-KB phrases, such as the terminologies in KP20k. Second, it can recognize false phrases just because they have similar surface names with real phrases. In the example from KPTimes, the model recognizes “*taxes companies*” and “*but companies*” as two phrases, while “*taxes*” is used as a verb in this sentence, and “*but*” is a conjunction word. Overall, the results generated by UCPhrase are more accurate. There is also an interesting case in the example from KPTimes, where RoBERTa and UCPhrase recognize “*pharmaceutical companies*” as a complete phrase, while Spacy and AutoPhrase think “*technology and*” is also part of the phrase. It is debatable which one is better: both results can contribute to a high-quality phrase vocabulary. In fact, even human annotators cannot achieve perfect agreement in their independent annotations. Dynamically adjusting the granularity of tagged phrases according to different end tasks remains a valuable research problem for further studies.

## 5 RELATED WORK

Phrase mining is a long studied task [6, 8, 11, 25, 34]. Due to the broad applicability of phrases to text-associated tasks, supervision signals would be expensive to obtain for vast domains. Unsupervised approaches have been proposed to extract phrases from many different angles, most importantly, language grammar [17, 27, 31] and text statistics [8]. Our work utilizes contextualized features from Transformer-based language models [7, 26], therefore, lifts the unnecessary requirement of frequency in statistics-based methods and alleviates requirements of expert-crafted grammar rules.

Table 7: Examples from silver training labels with extremely high and low quality scores  $f(\cdot : \theta)$  estimated by UCPhrase. The results show that UCPhrase is robust to noises in training labels.

$f(\cdot : \theta) > 99\%$	KP20k	$f(\cdot : \theta) < 1\%$
model identification, data structures, release dates, VLSI design, product development, network flow, finite precision, watermark detection, model selection, path planning, network security, data centers, source code, ...		times fewer, prescriptions implies, algorithms require estimating, significantly improves performance, including discontinuities, significantly reduce power consumption, factors include, <i>considered by</i> Titterton ...
$f(\cdot : \theta) > 99\%$	KPTimes	$f(\cdot : \theta) < 1\%$
Davis Cup, Ivy League, no-fly zone, Tour Championship, tax returns, City Hall, home runs, detention center, operating system, Ryder Cup, space stations, ice packs, White House, Jersey City, board games, tax cuts, ...		<i>PThe percentage</i> , 11th title, departments began telling officers, category includes workers, attacks including, 74th career win, including political, countries including Spain, including banking, including mobile, ...

Through experiments of three different tasks (i.e., corpus-level phrase ranking, document-level keyphrase extraction, and sentence-level phrase tagging), our method shows great performance improvement over previous methods.

Another line of research studies on distant supervision signals, such as existing knowledge bases [34, 36]. They typically use knowledge base entries (e.g., Wiki Entities from [34]) to string-match a corpus to obtain supervision signals in their first step. Such matching does not take into account how n-grams exist in the corpus, and as we show, could lead to partial matching of phrases, thus bringing bias to the phrase mining tool trained (e.g., “*heat island effect*” is usually matched into “*island effect*”). Our core phrase mining method, while being unsupervised, looks into the context of each n-gram to find max patterns and is able to find more complete phrases that serve as a better supervision signal to UCPhrase contextualized feature based classifier.

We use attention maps from pre-trained Transformer-based language models to identify phrases since they carry inter-relation information of tokens [4, 19]. Clark et al. [4] showed that a sufficient amount of linguistic knowledge, such as noun determiners and objects of verbs and prepositions, are captured by attention maps of BERT. Moreover, using only attention maps, one can train a model to perform dependency parsing [4] and constituency tree construction [19] relatively well. Our work utilizes this powerful nature of attention maps and treats them as the only feature to identify quality phrases. Furthermore, through comparing with the output states of RoBERTa, we show that using attention is less likely to overfit and has a more robust generalization.

## 6 CONCLUSIONS

We explore phrase tagging in an unsupervised and context-aware manner. Our proposed method, UCPhrase, shows clear improvement on performance for three quality-measuring tasks on two datasets in different domains. Further experimental studies reveal the strength of our two major components: our unsupervised core phrase mining finds more diverse, complete phrases in context than string-matching from some knowledge bases; our use of attention features unleashes the rich linguistic knowledge contained in pre-trained neural language models. By leveraging surface-agnostic context features, our model removes the frequency requirement in statistics-based models and alleviates the overfitting issue in embedding-based models.

We plan to explore the following directions in future studies. First, our study shows that the combination of silver labels and attention is robust and contains sufficient linguistic knowledge. This idea of unsupervised learning is worth exploring in other text mining tasks, such as coreference resolution [28], dependency parsing [21], and named entity recognition [30]. Second, the imperfection of distant supervision calls for a more effective way to incorporate large-scale unlabeled corpus with existing knowledge bases for more accurate prediction and more intelligent reasoning.

## ACKNOWLEDGMENTS

Research was supported in part by US DARPA KAIROS Program No. FA8750-19-2-1004, SocialSim Program No. W911NF-17-C-0099, National Science Foundation IIS-19-56151, IIS-17-41317, and IIS 17-04532, NSF Convergence Accelerator under award OIA-2040727, and the Molecule Maker Lab Institute: An AI Research Institutes program supported by NSF under Award No. 2019897.

## REFERENCES

- [1] Berna Altinel and Murat Can Ganiz. 2018. Semantic text classification: A survey of past and recent advances. *Information Processing & Management* 54, 6 (2018).
- [2] Steven Bird. 2006. NLTK: the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*. 69–72.
- [3] Florian Boudin. 2016. PKE: an open source python-based keyphrase extraction toolkit. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*. 69–73.
- [4] Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. 2019. What Does BERT Look at? An Analysis of BERT’s Attention. In *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*. 276–286.
- [5] W Bruce Croft, Howard R Turtle, and David D Lewis. 1991. The use of phrases and structured queries in information retrieval. In *Proceedings of the 14th annual international ACM SIGIR conference on Research and development in information retrieval*. 32–45.
- [6] Paul Deane. 2005. A nonparametric method for extraction of candidate phrasal terms. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL’05)*. 605–613.
- [7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. 4171–4186.
- [8] Ahmed El-Kishky, Yanglei Song, Chi Wang, Clare R Voss, and Jiawei Han. 2014. Scalable Topical Phrase Mining from Text Corpora. *Proceedings of the VLDB Endowment* 8, 3 (2014).
- [9] David A Evans and Chengxiang Zhai. 1996. Noun-phrase analysis in unrestricted text for information retrieval. In *Proceedings of the 34th annual meeting on Association for Computational Linguistics*. 17–24.
- [10] Geoffrey Finch. 2016. *Linguistic terms and concepts*. Macmillan International Higher Education.
- [11] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multi-word terms: the c-value/nc-value method. *International journal on digital libraries* 3, 2 (2000), 115–130.
- [12] Ygor Gallina, Florian Boudin, and Béatrice Daille. 2019. KPTime: A Large-Scale Dataset for Keyphrase Generation on News Documents. In *Proceedings of the 12th International Conference on Natural Language Generation*. 130–135.
- [13] Ygor Gallina, Florian Boudin, and Béatrice Daille. 2020. Large-scale evaluation of keyphrase extraction models. In *Proceedings of the ACM/IEEE Joint Conference on Digital Libraries in 2020*. 271–278.
- [14] Linyuan Gong, Di He, Zhuohan Li, Tao Qin, Liwei Wang, and Tieyan Liu. 2019. Efficient training of bert by progressively stacking. In *International Conference on Machine Learning*. PMLR, 2337–2346.
- [15] Xiaotao Gu, Linyuan Liu, Hongkun Yu, Jing Li, Chen Chen, and Jiawei Han. 2020. On the Transformer Growth for Progressive BERT Training. *arXiv preprint arXiv:2010.12562* (2020).
- [16] Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A Smith. 2020. Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. 8342–8360.
- [17] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. 2020. *spaCy: Industrial-strength Natural Language Processing in Python*.
- [18] Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991* (2015).
- [19] Taek Kim, Jihun Choi, Daniel Edmiston, and Sang-goo Lee. 2019. Are Pre-trained Language Models Aware of Phrases? Simple but Strong Baselines for Grammar Induction. In *International Conference on Learning Representations*.
- [20] Diederik P Kingma and Jimmy Ba. 2015. Adam: A Method for Stochastic Optimization. In *ICLR (Poster)*.
- [21] Sandra Kübler, Ryan McDonald, and Joakim Nivre. 2009. Dependency parsing. *Synthesis lectures on human language technologies* 1, 1 (2009), 1–127.
- [22] John Lafferty, Andrew McCallum, and Fernando CN Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. (2001).
- [23] Bing Li, Xiaochun Yang, Bin Wang, and Wei Cui. 2017. Efficiently Mining High Quality Phrases from Texts. In *AAAI*.
- [24] Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1054–1064.
- [25] 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*. 1729–1744.
- [26] 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. (2019).
- [27] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations*. 55–60.
- [28] Joseph F McCarthy and Wendy G Lehnert. 1995. Using decision trees for conference resolution. In *Proceedings of the 14th international joint conference on Artificial intelligence-Volume 2*. 1050–1055.
- [29] Rui Meng, Sanqiang Zhao, Shuguang Han, Daqing He, Peter Brusilovsky, and Yu Chi. 2017. Deep Keyphrase Generation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 582–592.
- [30] David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Linguisticae Investigationes* 30, 1 (2007), 3–26.
- [31] Graham Neubig, Taro Watanabe, Eiichiro Sumita, Shinsuke Mori, and Tatsuya Kawahara. 2011. An unsupervised model for joint phrase alignment and extraction. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*. 632–641.
- [32] Lance A Ramshaw and Mitchell P Marcus. 1999. Text chunking using transformation-based learning. In *Natural language processing using very large corpora*. Springer, 157–176.
- [33] Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*. 147–155.
- [34] 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 (2018), 1825–1837.
- [35] Jingbo Shang, Linyuan Liu, Xiaotao Gu, Xiang Ren, Teng Ren, and Jiawei Han. 2018. Learning Named Entity Tagger using Domain-Specific Dictionary. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*.
- [36] Li Wang, Wei Zhu, Sihang Jiang, Sheng Zhang, Keqiang Wang, Yuan Ni, Guotong Xie, and Yanghua Xiao. 2020. Mining Infrequent High-Quality Phrases from Domain-Specific Corpora. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*. 1535–1544.
- [37] Jake Ryland Williams, Paul R Lessard, Suma Desu, Eric M Clark, James P Bagrow, Christopher M Danforth, and Peter Sheridan Dodds. 2015. Zipf’s law holds for phrases, not words. *Scientific reports* 5, 1 (2015), 1–7.