# Understanding the Tradeoff between Cost and Quality of Expert Annotations for Keyphrase Extraction

**Hung Chau***
University of Pittsburgh
Pittsburgh, PA
hkc6@pitt.edu

**Saeid Balaneshin**
Zillow Group
Seattle, WA
saeidb@zillowgroup.com

**Kai Liu**
Zillow Group
Seattle, WA
kail@zillow.com

**Ondrej Linda**
Zillow Group
Seattle, WA
ondrejl@zillow.com

## Abstract

Generating expert ground truth annotations of documents can be a very expensive process. However, such annotations are essential for training domain-specific keyphrase extraction models, especially when utilizing data-intensive deep learning models in unique domains such as real-estate. Therefore, it is critical to optimize the manual annotation process to maximize the quality of the annotations while minimizing the cost of manual labor. To address this need, we explore multiple annotation strategies including self-review and peer-review as well as various methods of resolving annotator disagreements. We evaluate these annotation strategies with respect to their cost and on the task of learning keyphrase extraction models applied with an experimental dataset in the real-estate domain. The results demonstrate that different annotation strategies should be considered depending on specific metrics such as precision and recall.

## 1 Introduction

Automatic keyphrase extraction is an important technology on the crossroads of natural language processing and information access. Domain-specific keyphrase extraction models are widely used in many real-world applications such as document characterization and clustering (Hammouda et al., 2005), domain specific knowledge organization (Kosovac et al., 2002), topic-based access to document collections (Jones and Paynter, 1999), natural language question answering (Chaudhri et al., 2013), personalized recommendation of external content (Agrawal et al., 2014), and many other tasks (Papagiannopoulou and Tsoumakas, 2019).

One approach to such keyphrase extraction is to apply pre-trained or unsupervised models. However, such models might suffer from a lack of domain-specific knowledge. For example, while the terms "home" or "bedroom" alone can be called keyphrases in general, they carry very little information in the real-estate domain. Therefore, training domain-specific keyphrase extraction models based on expert knowledge is vital in such applications. In addition, for some easier tasks such as content linking or content recommendation, automatic processing could support sufficient levels of quality. For more challenging tasks, such as personalization, the use of expert annotation in some form is essential.

Annotations of domain-specific documents can be performed via crowd-sourcing, online workers and/or expert annotators (Su et al., 2007; Snow et al., 2008). This process can be done by each expert annotating a single document, doing self-review, multiple experts annotating the same document or doing peer-reviews. In the case of multi-annotator disagreement, a voting rule should be applied. In previous studies, a common practice to measure the quality of annotation labels is to compute inter-annotator agreement (Wilbur et al., 2006; Ogren et al., 2006; Kim et al., 2008; South et al., 2014; Augenstein et al., 2017). More annotation/review steps involved in this process often result in a better agreement between annotators.

* The author completed this work during an internship at Zillow Group.





*The 14th Linguistic Annotation Workshop*, pages 74–86
Barcelona, Spain, December 12, 2020.

However, it is still unclear whether the training set extracted from annotations with multiple annotation/review steps would increase the performance of keyphrase extraction models significantly. In addition, multiple annotation/review steps inevitably increase the cost of the annotation process as multiple experts have to tag the same set of documents. Due to the cost of hiring domain experts, it is critical to select an annotation strategy that maximizes the quality while minimizing the cost. Despite this substantial higher cost, to the best of our knowledge, there is no study analyzing the tradeoff of using annotations by multiple experts and the performance improvements of the keyphrase extraction model.

To fill in this gap, this paper experimentally analyzes the tradeoff between the cost of annotation strategies and the impact these strategies have on the model performance. We measure the mean annotation time per document and compare the precision, recall and f1-score measures of classification and sequence labeling models over self-review, peer-review and different annotation aggregation methods. The experimental results demonstrate that different annotation strategies should be selected depending on whether the objective is to optimize for precision, recall or f1-score.

The rest of the paper is organized as follows. Section 2 reviews related work and Section 3 discusses annotation and keyphrase extraction methodology. The experimental results are presented in 4 and the paper is concluded in 5.

## 2 Related Work

### 2.1 Data Annotation

Annotation is the basis of any supervised natural language processing research. Annotation processes has been applied in various domains, including the scientific publication domain (Augenstein et al., 2017), biomedical literature (Wilbur et al., 2006; Kim et al., 2008), educational textbooks (Chau et al., 2020), and medical records (Ogren et al., 2006; Xia and Yetisgen-Yildiz, 2012). These processes often start with recruiting domain experts and defining initial guidelines. Next, the guidelines are iteratively refined until the agreement reaches a pre-defined threshold. Each expert then annotates a larger scale document collection using the guideline. Eventually, ground truth data is selected based on the inter-annotator agreement. In the annotation process, organizing meetings with the annotators and involving them in reviewing and adjudication have shown positive impacts on the quality of annotations (Kim et al., 2008; Chau et al., 2020) but of course increase annotation efforts. Interestingly, Wilbur et al. (2006) found that the inter-annotator agreement could significantly increase among annotators who had gained experience working with the guidelines.

There are efforts in exploring the possibility of utilizing crowdsourcing for domain-specific text annotations. For instance, Uzuner et al. (2010) addressed the problem of annotating documents in the medical domain through experts to generate guidelines and a community of medical practitioners to perform annotations, which has a lower cost in comparison to employing experts for annotations. By utilizing guidelines generated by the experts and distributing them to the community annotators, they have achieved promising annotations. This indicates the value of including knowledge of the domain in generating high-quality annotation results in domain-specific tasks. Sabou et al. (2014) proposed a set of best practice guidelines for crowdsourced corpus acquisition and introduced an extension of the GATE NLP platform to facilitate the creation of crowdsourced tasks based on best practice. Liu et al. (2016) found that crowdsourced annotation can boost F1 score in relation extraction by Gated Instruction, which combines an interactive tutorial feedback to correct errors during training and improved screening, and they also claimed that with the high quality Gated Instruction annotations, a single annotation is more effective than majority vote over multiple annotators.

In contrast, researchers also explore the application of machine-assisted methods in expediting the text annotation process. A web survey conducted in 2009 (Tomanek and Olsson, 2009) shows that 20% of participants said they had used active learning as support in their annotation projects. There are also tools enabling semi-automatic annotation process, such as TURKSENT (Eryiğit et al., 2013), BRAT (Stenetorp et al., 2012), eHOST (South et al., 2012), and NER (Chen et al., 2017). They can generate annotations via experts correcting the output of a pre-trained linguistic system. Stenetorp et al. (2012) found a 15% decrease in total annotation time for a multicategory entity mention annotation task. However, South

et al. (2014) claimed that manual annotation process produced higher quality data without taking more time in comparison with an annotation method that combines machine pre-annotations with an interactive annotation interface in the manual annotation process.

For labeling keyphrases in text documents, the process of gathering the annotators (usually weekly) to discuss, resolve conflicts and agree on the annotations is very expensive. Allowing annotators to review their own or others's annotation may help to improve annotation quality with lower cost. However, it is still not clear how those extra efforts could help to increase keyphrase extraction models overall. This study attempts to understand the tradeoff between those cost and quality of the keyphrase annotation.

### 2.2 Keyphrase Extraction

There is a wide range of automatic keyphrase extraction methods from using unsupervised learning to rule-based, supervised learning or deep neural network models. Typical keyphrase extraction systems firstly pre-process data, extract *candidate keyphrases* using predefined Lexico-Syntactic patterns (Florescu and Caragea, 2017; Le et al., 2016), Part-of-Speech (POS) tags (e.g., *nouns* or *noun-nouns*) (Mihalcea and Tarau, 2004; Bougouin et al., 2013; Liu et al., 2009a; Wan and Xiao, 2008) or $n$-grams with simple filtering rules (Witten et al., 1999; Medelyan et al., 2009); and then predict which of these candidates are correct keyphrases.

An example of *unsupervised* keyphrase extraction methods are *graph-based methods* explored by (Mihalcea and Tarau, 2004; Bougouin et al., 2013). They consider a candidate keyphrase as important if it is related to a large number of candidates and those candidates are also important in the document. Candidates and their relations form a graph for the given document and keyphrases are selected based on their *PageRank* score. In addition, *topic-based clustering methods* (Liu et al., 2009b; Liu et al., 2010; Grineva et al., 2009) attempt to group semantically similar candidates in a document as *topics*. Keyphrases are then selected based on the centroid of each cluster or the importance of each topic. Although unsupervised learning models can extract keyphrases without any need for labeled data, their performances are commonly insufficient.

*Supervised* keyphrase extraction models often frame this task as *binary classification* or *sequence labeling problems*. The classifiers use different kinds of features, including *statistics*-based features, *linguistics*-based features or *external resources* (Hammouda et al., 2005; Witten et al., 1999; Rose et al., 2010; Hulth, 2003; Wang et al., 2015; Yih et al., 2006; Nguyen and Kan, 2007; Chau et al., 2020) to train supervised models. Sequence labeling models for keyphrase extraction have shown promising results in a recent study (Gollapalli et al., 2017). The deep sequence labeling with Bi-LSTM-CRF models has shown to significantly outperform its unsupervised and supervised baseline models (Alzaidy et al., 2019). However, the deep learning models require a large amount of data to achieve their best performances compared with traditional machine learning approaches.

Due to the high cost of creating training data, advanced weak supervision approaches have recently been attractive to the NLP community; however, expert knowledge is still needed to define labeling functions (especially in specific domains) and extra steps are usually applied to create cleaner training data outputs for ML models (e.g., slice-based learning) (Ratner et al., 2017; Chen et al., 2019). In this study, we focus on understanding the cost and quality of expert annotation for keyphrase extraction when manual annotation is essential, and compare supervised models to unsupervised models which do not need extra efforts for the problem.

## 3 Methodology

### 3.1 Annotation Procedure

To evaluate the tradeoff between cost and quality of expert annotation labels, we analyze multiple methods and aggregation strategies (voting rules) for label generation. In this annotation process, three experts, who have at least six months experience of working in the real-estate domain, receive training and pass a test that focuses on the understanding of the task and the BRAT annotation interface[1]. They, then, develop guidelines (described in Section 3.1.1) through multiple weekly annotation discussions. In each iteration,

[1] https://brat.nlplab.org

| Guidelines |
|---|
| Keyphrases should not contain multiple pieces of information. |
| Keyphrases should be the longest consecutive phrases. |
| Keyphrases should not have redundant words. |
| Keyphrases can have misspellings. |

Table 1: An example set of guidelines created by the experts.

the experts independently label keyphrases in 10 listing descriptions, followed up with a discussion to resolve disagreement and updating the guidelines. Next, the experts annotate a larger set of 50 listing descriptions and record the annotation time. The experts are not allowed to discuss their annotation/review before finishing all annotation tasks. All labels generated in the procedure are combined based on different strategies (described in Section 3.1.2) and used as ground truth data for training classification and sequence labeling models.

### 3.1.1 Guidelines

The annotation guidelines are created by experts through multiple annotation-discussion iterations. These iterations should continue until no guidelines are added or updated by the experts in a discussion session. An example set of these guidelines is shown in Table 1.

Some of the created guidelines can be used in the review process (presented in the next section) to reject an annotated keyphrase. For example, the guideline "*Keyphrases should not contain multiple pieces of information*" can be used to reject "4 bedrooms near Disneyland" which contains two pieces of information: "4 bedrooms" and "near Disneyland". On the other hand, a number of created guidelines such as "*Keyphrases can have misspellings*" can be used to guide the experts to accept keyphrases like "4 bedrom". For the guideline "*Keyphrases should be the longest consecutive phrases*", *3 bedrooms* is the keyphrase in "This home has 3 bedrooms." but *3 bedrooms upstairs* is the keyphrase in "It has 3 bedrooms upstairs".

The experts may combine guidelines to select or reject a keyphrase. For example, an expert may consider the guideline "*Keyphrases should not contain multiple concepts*" in conjunction with the guideline "*Keyphrases should be the longest consecutive phrases*" and "*Keyphrases should not have redundant words*" to select/reject a keyphrase.

We use the annotation labels generated during the guideline development process as ground truth data for model evaluation during our experiments.

### 3.1.2 Coding Procedure for Training Set

Having experts resolve annotation conflicts via discussion can be very expensive and not scalable for a large number of listing descriptions. Instead, a review process can be adopted to ensure the keyphrases adhere to the guidelines. Performing review after annotation steps increases the cost of annotation, but it may help to improve the quality of the keyphrase labels by mitigating issues such as experts' fatigue and lack of attention. An additional factor influencing the tradeoff between annotation cost and quality is the number of experts required to annotate each listing.

The review process can be either a self-review or a peer-review process. In the case of a self-review process, the same expert reviews his/her keyphrase annotation to ensure that he/she has followed the provided guidelines. The self-review also provides a chance for the experts to adjust their annotations based on their interpretations of guidelines so far. For a peer-review, the experts are exposed to the interpretation of the guidelines by the other experts. This exposure may cause the experts to change their mind about some of their interpretations of guidelines. The final annotations are the results of the reviewers' edits.

By considering the mean annotation time per document including the review process, the lowest cost approach is to have each document annotated by a single expert with no review. On the other hand, the most costly approach is to annotate the same listing descriptions by all the experts and perform a peer-review. This process is schematically depicted in Figure 1.

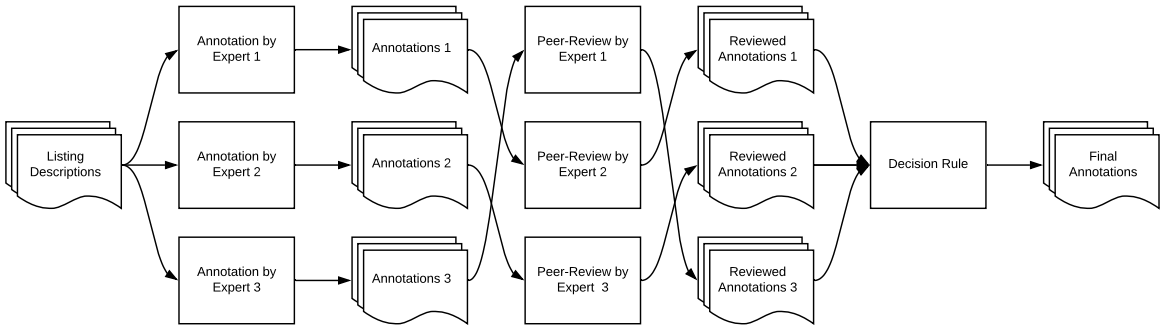



Figure 1: Process of generating training data based on peer-review.

### 3.2 Keyphrase Extraction Models

A domain-based keyphrase extraction task can be formulated as follows: *given a domain-specific text document, extract all phrases of interest to the users of that domain.* For this task, we investigate a shallow classification model and a deep sequence labeling model.

#### 3.2.1 Classification Model (LogReg)

For the classification model, we recast the keyphrase extraction as a binary classification problem including three main steps:

**Pre-processing data**: We use spaCy[2], an open-source library for NLP, to tag part-of-speech (POS) and then apply pre-defined patterns with regular expressions to extract all possible candidates (i.e., mostly but not limited to nouns and noun phrases). We only extract keyphrase candidates which consist of a maximum of 4 words in accordance with our annotation guidelines.

**Feature Extraction**: we extract an extensive list of features for each of the candidates:

- *linguistic*-based features: length of $n$-grams, concatenated POS of all tokens (e.g., `["JJ", "NN"]` for "great location"), POS of each of the tokens, POS of two words before, POS of two words after, and whether the phrase contains any named entities (e.g., area names).
- *statistics*-based features: document term frequency, collection term frequency, tf-idf, okapi BM25 and c-value (i.e., calculated from a collection of five thousand listing descriptions).

We bin and discretize non-binary numerical features in our model. We also apply a one-hot encoding on all non-binary features.

**Model training and prediction**: A logistic regression model (LogReg) is trained on the labeled feature vectors of candidate keyphrases. For the prediction phase, an input document is also processed by the first two steps and then the trained model will predict keyphrase likelihood for all candidates.

#### 3.2.2 Sequence Labeling Model (Bi-LSTM-CRF)

We also approach the keyphrase extraction problem as a sequence labeling task. This task can be formally stated as a named entity recognition (NER) problem. Given a sequence of *n* words in a listing description $d = \{w_1, w_2, ..., w_n\}$, we want to infer their hidden class labels (i.e., belonging to a keyphrase class). In this model, we set 3 class labels $Y = \{k_B, k_I, k_O\}$, representing "beginning of a keyphrase", "inside of a keyphrase", and "not a part of a keyphrase".

In this study, we apply a Bi-LSTM-CRF architecture to perform this task, which has been shown to achieve the best performance across several public datasets (Alzaidy et al., 2019). The standard Bi-LSTM-CRF model consists of three main components (see Figure 2). We briefly present these components as below, for the detailed architecture refer to this work (Liu et al., 2018).

**Embedding Layer**: word and character-level embeddings are trained purely on un-annotated sequence data from a text corpus. While word embeddings capture syntactic and semantic regularities in language, character embeddings provide additional information about the underlying style and structure of words,

[2] `https://spacy.io/`

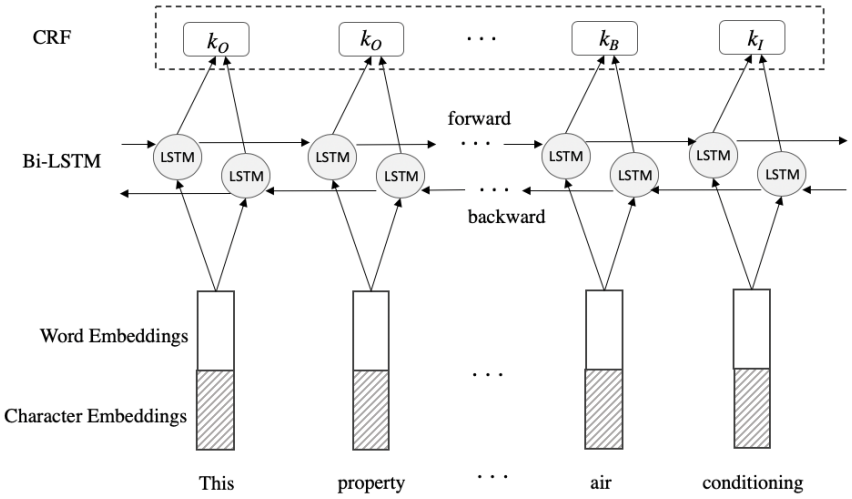



Figure 2: Bi-LSTM-CRF model for keyphrase extraction.

both improving many NLP tasks including NER. The one-hot vector of an input word $w_t$ is mapped to a fixed size dense vector in this layer.

**Bi-LSTM Layer**: the concatenation of character and word embeddings is the input for this layer. An LSTM unit has four components: *input gate*, *forget gate*, *memory cell* and *output gate*. The input vectors go through LSTM units in both directions, creating two hidden state vectors: $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ capturing information from words before and after $w_t$, respectively. The concatenation of these two vectors $\overleftrightarrow{h_t}$ represents the semantics and dependencies of $w_t$ in the context of the input text.

**CRF Layer**: Conditional Random Field (CRF) based models introduced by (Lafferty et al., 2001) have been successfully used in many sequence labeling tasks. $\overleftrightarrow{h_t}$ is the input for the CRF layer which produces a probability distribution over a tag sequence based on the mapping of the input vectors to the class space and the dependencies of adjacency class labels of the entire sequence. CRFs use the Viterbi algorithm to efficiently infer the optimal sequence of labels for an input sequence.

Our implementation of the model is based on the version presented in (Liu et al., 2018)[3]. We use the Glove pre-trained word embeddings of 100-dimensions[4]. Character embeddings are trained along with the main model with Bi-LSTM networks. The dimension of character embeddings is set to 30. We use a 300-dimension hidden layer for the character learning model as well as the main model. The models are trained using mini-batch stochastic gradient descent with momentum. The batch size is set to 5. The learning rate and decay ratio are set to 0.015 and 0.05, respectively. Dropout and gradient clipping of 5.0 are also applied to avoid over-fitting and increase stability.

## 4 Experiments and Results

### 4.1 Annotation Data Analysis

**Statistics**: We focus on the real-estate domain in English and create a dataset with 50 and 20 listing descriptions for training and evaluation. The average length of the sampled listing descriptions was 125 words, with some having as few as 50 and as many as 500. Table 2a lists the mean annotation time for each expert. It shows that, on average, experts conducted self- and peer-review in about half the time as the initial annotation.

**Count of Keyphrases**: Table 2b shows the average count of keyphrases per listing selected by experts in different steps of the annotation/review process. This table indicates that, on average, self- and peer-review steps slightly increase the number of selected keyphrases, indicating that experts more often added additional keyphrases than removed the already annotated ones.

[3] https://github.com/LiyuanLucasLiu/LM-LSTM-CRF

[4] https://nlp.stanford.edu/projects/glove/

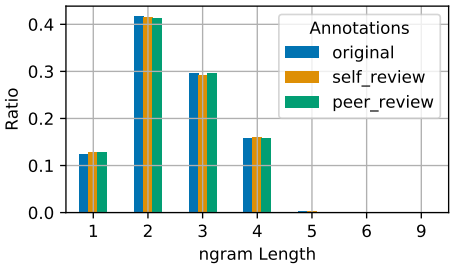
| Step | Annotation | Self-Review | Peer-Review |
|---|---|---|---|
| Time | 00:02:49 | 00:01:23 | 00:01:25 |

(a)

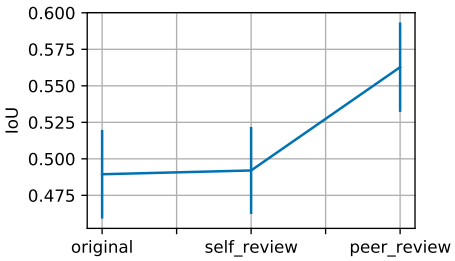
| Step/Expert | A | B | C |
|---|---|---|---|
| Annotation | 20.52 | 18.52 | 20.8 |
| Self-Review | 20.34 | 19.66 | 22.62 |
| Peer-Review | 20.74 | 20.24 | 21.98 |

(b)

Table 2: (a) Average time spent (in HH:MM:SS format) by each expert on each annotation/review step. (b) The average count of keyphrases selected by different experts from each listing description.




(a)




(b)

Figure 3: (a) The ratio of keyphrases with different lengths of $n$-grams. Bigrams are the most selected keyphrases. (b) Agreement (in terms of IoU) in selected keyphrases by different experts in different annotation/review steps. Peer-review is more effective than self-review in increasing the agreement.

**$n$-grams in Selected Keyphrases**: Figure 3a shows that the majority of selected keyphrases were bigrams (around 40%). Although per the guidelines, the experts were asked to limit the length of the annotated $n$-grams to 4, in less than 1% of cases, 5- and 6-grams were selected due to either a mistake or an incorrect interpretation of the guidelines (e.g., by selecting "14 x 14 covered dec" as a keyphrase). Figure 3a also shows that the length of the selected keyphrase did not change significantly after self- or peer-review processes.

**Self- and Peer-Review Processes**: In Figure 3b, we use the intersection-over-union (IoU) to measure the amount of agreement among experts, which is defined as # of keyphrases selected by all experts over # of keyphrases selected by any expert. The value of IoU ranges from 0 to 1, where 0 and 1 corresponds to no common keyphrases and all keyphrases being shared between experts, respectively.

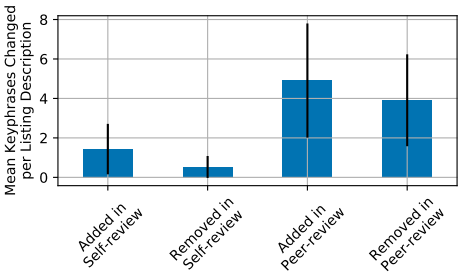



Figure 4: The average number of keyphrases added/removed per listing description during self and peer review steps.

| Method | Precision | Recall | F1-score |
|---|---|---|---|
| TextRank | 0.32 | 0.46 | 0.35 |
| SingleRank | 0.36 | 0.46 | 0.39 |
| TopicRank | 0.28 | 0.47 | 0.33 |
| TopicalPageRank | 0.32 | 0.41 | 0.35 |
| PositionRank | 0.31 | 0.44 | 0.34 |
| MultipartiteRank | 0.28 | 0.51 | 0.34 |
| LogReg | **0.59** | **0.83** | **0.69** |
| Bi-LSTM-CRF | **0.60** | **0.66** | **0.63** |

Table 3: The performances of two supervised models (LogReg and Bi-LSTM-CRF) trained on *orig-one* dataset in comparison to multiple unsupervised baselines.

| Dataset | Average no. of keyphrases per Doc. | Time per Doc. (sec) | LogReg | | | Bi-LSTM-CRF | | |
|---|---|---|---|---|---|---|---|---|
| | | | Precision | Recall | F1 | Precision | Recall | F1 |
| orig-one | 19.66 | 170 | 0.59 | 0.83 | 0.69 | 0.60 | 0.66 | 0.63 |
| orig-two-union | 24.3 | 340 | 0.54 | 0.86 | 0.66 | 0.59 | 0.68 | 0.63 |
| orig-two-unanimity | 15.34 | 340 | 0.60 | 0.83 | 0.69 | 0.70 | 0.58 | 0.64 |
| orig-three-union | 26.88 | 510 | 0.51 | 0.87 | 0.65 | 0.57 | 0.70 | 0.63 |
| orig-three-majority | 19 | 510 | 0.58 | 0.83 | 0.68 | 0.63 | 0.68 | 0.65 |
| orig-three-unanimity | 13.28 | 510 | **0.61** | 0.80 | 0.69 | **0.74** | 0.56 | 0.64 |
| self-one | 20.84 | 255 | 0.58 | 0.86 | 0.69 | 0.59 | 0.68 | 0.63 |
| self-two-union | 25.42 | 510 | 0.52 | 0.87 | 0.65 | 0.58 | 0.69 | 0.63 |
| self-two-unanimity | 15.98 | 510 | 0.60 | 0.84 | **0.70** | 0.68 | 0.62 | 0.65 |
| self-three-union | 28.1 | 765 | 0.49 | 0.87 | 0.63 | 0.52 | 0.72 | 0.60 |
| self-three-majority | 19.84 | 765 | 0.56 | 0.83 | 0.67 | 0.61 | 0.71 | 0.66 |
| self-three-unanimity | 13.94 | 765 | **0.61** | 0.82 | **0.70** | 0.71 | 0.59 | 0.64 |
| peer-one | 20.72 | 253 | 0.55 | 0.83 | 0.67 | 0.63 | 0.69 | 0.66 |
| peer-two-union | 24.86 | 506 | 0.51 | **0.89** | 0.65 | 0.56 | **0.74** | 0.64 |
| peer-two-unanimity | 16.7 | 506 | 0.60 | 0.85 | **0.70** | 0.66 | 0.66 | 0.66 |
| peer-three-union | 26.82 | 759 | 0.51 | **0.89** | 0.65 | 0.54 | **0.74** | 0.63 |
| peer-three-majority | 20.22 | 759 | 0.55 | 0.84 | 0.67 | 0.64 | 0.70 | **0.67** |
| peer-three-unanimity | 15.12 | 759 | 0.59 | 0.82 | 0.69 | 0.66 | 0.62 | 0.64 |

Table 4: The performance comparison of LogReg and Bi-LSTM-CRF models trained on different variations of our dataset.

Figure 3b shows that the ratio of common keyphrases between experts significantly increases during the peer-review process, while it remains unchanged during self-review. Therefore, in resolving the disagreements among experts, peer-review appears to be far more effective than self-review. The increase in agreements after the peer-review process can stem from the way that experts reconsidered their interpretation of guidelines when they were exposed to the annotations by other experts.

Top keyphrases that experts considered as acceptable after the peer-review process include "Dock", "Deck", and "Conveniently Located", and those removed after the peer-review process include "Stunning" and "Amenities".

When examining the average number of added and removed keyphrases across all listing descriptions, we found that the number of added/removed keyphrases in the peer-review process is around 5 times more than that in self-review process and that the number of added keyphrases is higher than removed in both self- and peer-review processes (see Figure 4). It suggests that experts more frequently tend to identify additional keyphrases that might have been missed previously in either self- or peer-review process.

### 4.2 Model Performance Comparison

As described in Section 3.1.2, we investigate the tradeoff between quality and cost of the annotations given the number of experts annotating each listing description and the type of review process. To do so, each listing description is firstly annotated by three experts separately and then it goes through the self- and peer-review processes. Therefore, the data we collected includes three annotations per each description. This allows us to create training sets with one or two expert annotations per listing description via uniformly sampling which one or two annotations out of the three available should be considered.

In this study, we also investigate three review types: (1) *orig*: original (no-review), (2) *self*: self-review, or (3) *peer*: peer-review. In the case of having more than one annotation per listing description, we investigate three voting rules: (1) *union*: a keyphrase is selected if it was annotated by at least one expert, (2) *majority*: a keyphrase is selected if it was annotated by two or more experts, or (3) *unanimity*: a keyphrase is selected if it was annotated by all three experts. By combining review types, number of annotators per listing description and the voting rules, we generated 18 different training data sets. The name of these data sets are described in Table 4 by the following format: review type (*orig*, *self*, or *peer*)-number of annotations per listing description (*one*, *two*, *three*)-voting rule (*union*, *majority*, or *unanimity*).

In Table 4, we show the performance of the classification (LogReg) and sequence labeling (Bi-LSTM-

| Output from LogReg | Output from Bi-LSTM-CRF |
|---|---|
| A ranch style home inside the Western Park community! Entering the home you step inside the large living room that basks in plenty of sunshine. The eat-in kitchen overlooks the living room, allowing you to chat while you cook dinner! The kitchen boasts countertop space, upper cabinetry, a large pantry, and sleek black appliances. | A ranch style home inside the Western Park community! Entering the home you step inside the large living room that basks in plenty of sunshine. The eat-in kitchen overlooks the living room, allowing you to chat while you cook dinner! The kitchen boasts countertop space, upper cabinetry, a large pantry, and sleek black appliances. |

Table 5: LogReg vs. Bi-LSTM-CRF: keyphrase extraction in real estate. The yellow keyphrases are true positives and the blue ones are false negatives.

CRF) keyphrase extraction models trained on all the 18 training data sets and evaluated on the common ground truth data, which as described in Section 3.1.1 includes 20 descriptions and has average of 14.8 keyphrases per document. As an example of the final output, Figure 5 depicts a listing description and the extracted phrases from LogReg and Bi-LSTM-CRF. In addition to comparing precision, recall and f1-scores, we also include the average number of keyphrases and the mean annotation time per listing description for each training set. The presented experimental results allow us to analyze the tradeoff between the cost and quality of the selected keyphrase annotation methods.

Observations from Table 4:

- **Precision vs. Time**: For both LogReg and Bi-LSTM-CRF models, using *orig-three-unanimity* data set results in the highest precision value.
- **Recall vs. Time**: the best recall was achieved by *peer-two-union* for both LogReg and Bi-LSTM-CRF models with regard to time.
- **F1-score vs. Time**: the best performance was achieved by *peer-two-unanimity* for LogReg and by *peer-three-majority* for Bi-LSTM-CRF models. However, in the case of LogReg model, *orig-one* only needs 170 seconds but its performance is very close to the best, which requires on average 506 seconds of annotation time.
- **Precision and Recall vs. Voting Rule**: From these results we can conclude that the more agreement is enforced among the annotators, the higher precision (e.g., Precision(**-three-unanimity*) $>$ Precision(**-three-majority*) $>$ Precision(**-three-union*)). The result consistently indicates that the larger the size of the training data, the higher the recall (e.g., Recall(**-three-union*) $>$ Recall(**-three-majority*) $>$ Recall(**-three-unanimity*)).
- **LogReg vs. Bi-LSTM-CRF**: the recall of LogReg model is higher than Bi-LSTM-CRF model; on the other hand, the precision of the latter is higher than the former. Overall, the F1-score of LogReg model is a bit better. The Bi-LSTM-CRF model typically requires much more labeled data to boost the performance or needs to fine tune on a pre-trained model. Nevertheless, with this small training set, the deep sequence labeling model is still able to obtain a good result that outperforms unsupervised models that will be presented shortly.
- ***Original* vs. *self-review* vs. *peer-review***: the average performance as well as the mean annotation time of the *self-review* and *peer-review* data are very similar. The *original* data, which requires least effort, has the lowest recall but surprisingly the highest precision. For the average F1-scores, we do not see significant differences among these three.

In Table 3, we compare the performances of the two supervised models with state-of-the-art unsupervised models (Mihalcea and Tarau, 2004; Wan and Xiao, 2008; Bougouin et al., 2013; Sterckx et al., 2015; Florescu and Caragea, 2017; Boudin, 2018).[5] We choose *orig-one* dataset, which requires the least annotation effort, as the training data for the supervised models. Table 3 reveals that the supervised models substantially outperform all the unsupervised baselines in terms of *precision*, *recall* and *f1-score*. This performance again emphasizes the importance of exploiting domain-specific knowledge and expert annotations as labeled training data for training keyphrase extraction models.

## 5 Conclusions

In this paper, we presented multiple annotation strategies including self-review and peer-review processes as well as various ways of resolving annotator disagreement for keyphrase annotation problems. We trained a classification model with an extensive list of features in the domain of real-estate. In addition,

[5] https://github.com/boudinfl/pke

we applied a Bi-LSTM-CRF architecture for a sequence labeling approach to extract keyphrases. We evaluated the two models' performances with eighteen different training datasets generated from the aforementioned strategies to see the tradeoff between the cost and quality of expert annotations. The results showed that different annotation strategies can be considered depending on a specific metric. We observed the consistent improvement for precision or recall when applying different voting rules for all the three review types. With respect to average f1-scores, we do not see an improvement of self-review and peer-review over the original annotations. The comparison between the two supervised models with the state-of-the-art unsupervised models has shown the importance of exploiting domain-specific knowledge and expert annotations to keyphrase extraction problems. However, this work is limited to one small dataset. It could be extended and evaluated on multiple datasets from different domains (e.g., e-commerce, education or medical) to examine the general applicability of our proposed annotation strategies.

This work, to the best of our knowledge, is the first to understand the tradeoff between cost and quality of expert annotation for keyphrase extraction. There is still room to improve the models, for example by leveraging user search terms as a feature for LogReg or pre-trained language models and transfer learning for Bi-LSTM-CRF. Our priority is to investigate whether the approach is valid for other NLP tasks such as *relation extraction*. We also plan to investigate how useful the generated code book for weak supervision modeling, how to translate the rules in the guidelines to labeling functions in weak supervision.

## References


Rakesh Agrawal, Sreenivas Gollapudi, Anitha Kannan, and Krishnaram Kenthapadi. 2014. Study navigator: An algorithmically generated aid for learning from electronic textbooks. *Journal of Educational Data Mining*, 6(1):53–75.

Rabah Alzaidy, Cornelia Caragea, and C. Lee Giles. 2019. Bi-LSTM-CRF sequence labeling for keyphrase extraction from scholarly documents. In *The World Wide Web Conference*, WWW '19, page 2551–2557, New York, NY, USA. Association for Computing Machinery.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017. Semeval 2017 task 10: ScienceIE-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*.

Florian Boudin. 2018. Unsupervised keyphrase extraction with multipartite graphs. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 667–672, New Orleans, Louisiana, June. Association for 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.

Hung Chau, Igor Labutov, Khushboo Thaker, Daqing He, and Peter Brusilovsky. 2020. Automatic concept extraction for domain and student modeling in adaptive textbooks. *International Journal of Artificial Intelligence in Education*.

Vinay K. Chaudhri, Britte Cheng, Adam Overtholtzer, Jeremy Roschelle, Aaron Spaulding, Peter Clark, Mark Greaves, and Dave Gunning. 2013. Inquire biology: A textbook that answers questions. *AI Magazine*, 34(3):55–72.

Yukun Chen, Thomas A Lask, Qiaozhu Mei, Qingxia Chen, Sungrim Moon, Jingqi Wang, Ky Nguyen, Tolulola Dawodu, Trevor Cohen, Joshua C Denny, et al. 2017. An active learning-enabled annotation system for clinical named entity recognition. *BMC medical informatics and decision making*, 17(2):35–44.

Vincent Chen, Sen Wu, Alexander J Ratner, Jen Weng, and Christopher Ré. 2019. Slice-based learning: A programming model for residual learning in critical data slices. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 9397–9407. Curran Associates, Inc.

Gülşen Eryiğit, Fatih Samet Cetin, Meltem Yanık, Tanel Temel, and Ilyas Çiçekli. 2013. Turksent: A sentiment annotation tool for social media. In *Proceedings of the 7th Linguistic Annotation Workshop and Interoperability with Discourse*, pages 131–134.

Corina Florescu and Cornelia Caragea. 2017. 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. Association for Computational Linguistics.

Sujatha Das Gollapalli, Xiao li Li, and Peng Yang. 2017. Incorporating expert knowledge into keyphrase extraction.

Maria Grineva, Maxim Grinev, and Dmitry Lizorkin. 2009. Extracting key terms from noisy and multitheme documents. In *Proceedings of the 18th International Conference on World Wide Web*, WWW '09, pages 661–670, New York, NY, USA. ACM.

Khaled M. Hammouda, Diego N. Matute, and Mohamed S. Kamel. 2005. Corephrase: Keyphrase extraction for document clustering. In Petra Perner and Atsushi Imiya, editors, *Machine Learning and Data Mining in Pattern Recognition*, pages 265–274, Berlin, Heidelberg. Springer Berlin Heidelberg.

Anette Hulth. 2003. Improved automatic keyword extraction given more linguistic knowledge. In *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing*, EMNLP '03, pages 216–223, Stroudsburg, PA, USA. Association for Computational Linguistics.

Steve Jones and Gordon Paynter. 1999. Topic-based browsing within a digital library using keyphrases. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, DL '99, page 114–121, New York, NY, USA. Association for Computing Machinery.

Jin-Dong Kim, Tomoko Ohta, and Jun'ichi Tsujii. 2008. Corpus annotation for mining biomedical events from literature. *BMC bioinformatics*, 9(1):10.

Branka Kosovac, Dana J. Vanier, and Thomas M. Froese. 2002. Use of keyphrase extraction software for creation of an aec/fm thesaurus. *Journal of Information Technology in Construction*, 5(2):25–36.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Tho Thi Ngoc Le, Minh Le Nguyen, and Akira Shimazu. 2016. Unsupervised keyphrase extraction: Introducing new kinds of words to keyphrases. In Byeong Ho Kang and Quan Bai, editors, *AI 2016: Advances in Artificial Intelligence*, pages 665–671, Cham. Springer International Publishing.

Feifan Liu, Deana Pennell, Fei Liu, and Yang Liu. 2009a. Unsupervised approaches for automatic keyword extraction using meeting transcripts. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 620–628, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhiyuan Liu, Peng Li, Yabin Zheng, and Maosong Sun. 2009b. Clustering to find exemplar terms for keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1 - Volume 1*, EMNLP '09, pages 257–266, Stroudsburg, PA, USA. Association for Computational Linguistics.

Zhiyuan Liu, Wenyi Huang, Yabin Zheng, and Maosong Sun. 2010. Automatic keyphrase extraction via topic decomposition. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, EMNLP '10, pages 366–376, Stroudsburg, PA, USA. Association for Computational Linguistics.

Angli Liu, Stephen Soderland, Jonathan Bragg, Christopher H. Lin, Xiao Ling, and Daniel S. Weld. 2016. Effective crowd annotation for relation extraction. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 897–906, San Diego, California, June. Association for Computational Linguistics.

Liyuan Liu, Jingbo Shang, Xiang Ren, Frank F. Xu, Huan Gui, Jian Peng, and Jiawei Han. 2018. Empower sequence labeling with task-aware neural language model. In *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 5253–5260. AAAI Press.

Olena Medelyan, Eibe Frank, and Ian H. Witten. 2009. Human-competitive tagging using automatic keyphrase extraction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3 - Volume 3*, EMNLP '09, pages 1318–1327, Stroudsburg, PA, USA. 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*, Barcelona, Spain, July.

Thuy Dung Nguyen and Min-Yen Kan. 2007. Keyphrase extraction in scientific publications. In Dion Hoe-Lian Goh, Tru Hoang Cao, Ingeborg Torvik Sølvberg, and Edie Rasmussen, editors, *Asian Digital Libraries. Looking Back 10 Years and Forging New Frontiers*, pages 317–326, Berlin, Heidelberg. Springer Berlin Heidelberg.

Philip V Ogren, Guergana Savova, James D Buntrock, and Christopher G Chute. 2006. Building and evaluating annotated corpora for medical nlp systems. In *AMIA Annual Symposium proceedings. AMIA Symposium*, volume 2006, pages 1050–1050. American Medical Informatics Association.

Eirini Papagiannopoulou and Grigorios Tsoumakas. 2019. A review of keyphrase extraction. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery (2019)*, e1339.

Alexander Ratner, Stephen H. Bach, Henry R. Ehrenberg, Jason Alan Fries, Sen Wu, and Christopher Ré. 2017. Snorkel: Rapid training data creation with weak supervision. *CoRR*, abs/1711.10160.

Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. Automatic keyword extraction from individual documents. In Michael W. Berry and Jacob Kogan, editors, *Text Mining: Applications and Theory*, pages 1–20. John Wiley and Sons, Ltd.

Marta Sabou, Kalina Bontcheva, Leon Derczynski, and Arno Scharl. 2014. Corpus annotation through crowdsourcing: Towards best practice guidelines. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 859–866, Reykjavik, Iceland, May. European Language Resources Association (ELRA).

Rion Snow, Brendan O'connor, Dan Jurafsky, and Andrew Y Ng. 2008. Cheap and fast–but is it good? evaluating non-expert annotations for natural language tasks. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 254–263.

Brett South, Shuying Shen, Jianwei Leng, Tyler Forbush, Scott DuVall, and Wendy Chapman. 2012. A prototype tool set to support machine-assisted annotation. In *BioNLP: Proceedings of the 2012 Workshop on Biomedical Natural Language Processing*, pages 130–139, Montréal, Canada, June. Association for Computational Linguistics.

Brett R South, Danielle Mowery, Ying Suo, Jianwei Leng, Oscar Ferrández, Stephane M Meystre, and Wendy W Chapman. 2014. Evaluating the effects of machine pre-annotation and an interactive annotation interface on manual de-identification of clinical text. *Journal of biomedical informatics*, 50:162–172.

Pontus Stenetorp, Sampo Pyysalo, Goran Topić, Tomoko Ohta, Sophia Ananiadou, and Jun'ichi Tsujii. 2012. BRAT: a web-based tool for NLP-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107, Avignon, France, April. Association for Computational Linguistics.

Lucas Sterckx, Thomas Demeester, Johannes Deleu, and Chris Develder. 2015. Topical word importance for fast keyphrase extraction. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, page 121–122, New York, NY, USA. Association for Computing Machinery.

Qi Su, Dmitry Pavlov, Jyh-Herng Chow, and Wendell C Baker. 2007. Internet-scale collection of human-reviewed data. In *Proceedings of the 16th international conference on World Wide Web*, pages 231–240.

Katrin Tomanek and Fredrik Olsson. 2009. A web survey on the use of active learning to support annotation of text data. *Proceedings of Active Learning for Natural Language Processing (ALNLP-09)*, pages 45–48.

Özlem Uzuner, Imre Solti, Fei Xia, and Eithon Cadag. 2010. Community annotation experiment for ground truth generation for the i2b2 medication challenge. *Journal of the American Medical Informatics Association*, 17(5):519–523.

Xiaojun Wan and Jianguo Xiao. 2008. Collabrank: Towards a collaborative approach to single-document keyphrase extraction. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 969–976. Coling 2008 Organizing Committee.

Shuting Wang, Chen Liang, Zhaohui Wu, Kyle Williams, Bart Pursel, Benjamin Brautigam, Sherwyn Saul, Hannah Williams, Kyle Bowen, and C. Lee Giles. 2015. Concept hierarchy extraction from textbooks. In *Proceedings of the 2015 ACM Symposium on Document Engineering*, DocEng '15, pages 147–156, New York, NY, USA. ACM.

W John Wilbur, Andrey Rzhetsky, and Hagit Shatkay. 2006. New directions in biomedical text annotation: definitions, guidelines and corpus construction. *BMC bioinformatics*, 7(1):1–10.

Ian H. Witten, Gordon W. Paynter, Eibe Frank, Carl Gutwin, and Craig G. Nevill-Manning. 1999. Kea: Practical automatic keyphrase extraction. In *Proceedings of the Fourth ACM Conference on Digital Libraries*, DL '99, pages 254–255, New York, NY, USA. ACM.

Fei Xia and Meliha Yetisgen-Yildiz. 2012. Clinical corpus annotation: challenges and strategies. In *Proceedings of the Third Workshop on Building and Evaluating Resources for Biomedical Text Mining (BioTxtM'2012) in conjunction with the International Conference on Language Resources and Evaluation (LREC), Istanbul, Turkey*.

Wen-tau Yih, Joshua Goodman, and Vitor R. Carvalho. 2006. Finding advertising keywords on web pages. In *Proceedings of the 15th International Conference on World Wide Web*, WWW '06, pages 213–222, New York, NY, USA. ACM.