HESDK: A Hybrid Approach to Extracting Scientific Domain Knowledge Entities

Jian Wu[†], Sagnik Ray Choudhury[*][†], Agnese Chiatti[†], Chen Liang[†], C. Lee Giles[†]

[†]Information Sciences and Technology, Pennsylvania State University, University Park, PA, 16802
{jxw394,szr163,azc76,cul226,giles}@ist.psu.edu

ABSTRACT

We investigate a variant of the problem of automatic keyphrase extraction from scientific documents which we define as *Scientific Domain Knowledge Entity* (SDKE) extraction. Keyphrases are noun phrases *important to the documents themselves.* In contrasxt, an SDKE is text that refers to a *concept* and can be classified as a process, material, task, dataset etc. A SDKE represents domain knowledge, but is not necessarily important to the document it is in. Supervised keyphrase extraction algorithms using non-sequential classifiers and global measures of informativeness (PMI, tf-idf) have been used for this task. Another approach is to use sequential labeling algorithms with local context from a sentence, as done in the named entity recognition. We show that these two methods can complement each other and a simple merging can improve the extraction accuracy by 5-7 percentiles. We further propose several heuristics to improve the extraction accuracy. Our preliminary experiments suggest that it is possible to improve the accuracy of the sequential learner itself by utilizing the predictions of the non-sequential model.

CCS CONCEPTS

•Information systems →Data mining; Digital libraries and archives; Information extraction; Content analysis and feature selection;

KEYWORDS

digital library, keyphrase extraction, conditional random field, natural language processing, scientific domain knowledge entity

ACM Reference format:
Jian Wu[†], Sagnik Ray Choudhury[*][†], Agnese Chiatti[†], Chen Liang[†], C. Lee Giles[†]
[†]Information Sciences and Technology, Pennsylvania State University, University Park, PA, 16802
{jxw394,szr163,azc76,cul226,giles}@ist.psu.edu
. 2017. HESDK: A Hybrid Approach to Extracting Scientific Domain Knowledge Entities. In *Proceedings of ACM/IEEE-CS Joint Conference on Digital Libraries, Toronto, Canada, June 2017 (JCDL'17),* 4 pages.
DOI:

[*]This author makes an equal contribution to this paper as the first author.

1 INTRODUCTION

Keywords and keyphrases are ubiquitous in digital library search engines. They provide a high-level topical description of documents and are very useful in data engineering, e.g., [4]. Considerable effort has been spent on automated methods for keyphrase extraction from scientific documents. Recent work includes unsupervised systems such as RAKE [8] and supervised systems such as Maui [7], and CeKE [1]. RAKE was tested on a dataset of 500 abstracts from journal papers in Computer Science and Information Technology. The ground truth of Maui was adopted from user tagged documents in CiteULike. CeKE used *author-input* keywords from a corpus of WWW and KDD conference proceedings.

Keyphrases provide a succinct summary of a document. But they lack concrete content descriptors of the text by which readers can have a better understanding of the *concepts* in a scientific document, such as materials, methods, tools, datasets, problems, and processes. As such, we define a new type of textual entity (phrase) for scientific documents which we name *Scientific Domain Knowledge Entity.* An SDKE represents domain knowledge. For example, in Computer Science papers, "Python", "Apache Solr", "Stanford CoreNLP", and "MySQL" are *tools*, "WebKB" and "DBLP" are *datasets*, "author name disambiguation", and "tree traversal" are *problems*; in Materials Science, "Alloy" and "Mg" are *materials.* Some of these can be keyphrases in the traditional sense, but these phrases provide a greater range of semantics. Consider a sentence from our training set:

> Each $\underline{\text{micro element}}$ is divided into $\underline{\text{spatially-oriented bins}}$ in the y-direction in order to $\underline{\text{resolve the velocity and shear-stress profiles}}$.

The underlined phrases are annotated as SDKEs.

We discriminate SDKEs from traditional keyphrases because they are different in both semantic density and what they represent. For example, the keyphrases used in [1] are assigned by authors. There are on average 4–5 keyphrases per document. The keyphrases used in [7] are tagged by individual users with about 26 keyphrases per document. The dataset we use in this study was adopted from SemEval 2017 challenge Task 10. This dataset was labeled by neither of the methods previously described. There are 6338 keyphrases identified among 350 documents in the training set, which is 18 keyphrases per document. Note that each document is just one paragraph of text between 60 and 264 words, so the labels are much denser. We adopted the SemEval 2017 annotation as the ground truth because it is the closest dataset for our purposes. Traditionally, keyphrases provide *top-level* description of the main ideas of a document. In contrast, SDKEs give a find-grained description of the document body.

The first two SDKEs in the example above are traditional keyphrases, i.e., important and informative noun phrases. The last one is not because it is not a noun phrase, and the word "velocity" is not important without considering in context the word "resolve" and "shear-stress". Thus we explore a hybrid approach where a combination of non-sequential and sequential classifiers can be integrated to extract relevant SDKEs. We believe these methods should be complimentary: the sequential classifier should capture the local dependence among the words while the non-sequential one should capture the global informativeness.

We use an unsupervised approach as a baseline to extract candidate phrases which are then filtered by a linear kernel SVM that utilizes global (calculated from the whole document) features. A linear-chain conditional random field[6] using word context and shallow syntactic features is used to learn the contextual dependence. As expected, these models learn different types of phrases. A simple merging of their results improves the extraction accuracy by 5-7%. Employing a set of heuristics we achieve a further increase of 5% in accuracy. Among the features used by the SVM, tf-idf turns out to be the most important and therefore is further used in the design of heuristics. While the task is modeled as extracting SDKEs from a paragraph of a paper and not the full text, we observe that the availability of the full text leads to better heuristics, improving the extraction accuracy. We also investigate whether one can use the SVM predictions directly to improve the accuracy of the CRF. While our initial resuls find no such evidence, we believe this should be further investigated.

We only focus on the *extraction* of the phrases and not the classification (inferring that the keyphrase is of type *Material, Process, Dataset*). While joint extraction and classification is indeed a more interesting problem, that is the scope of another paper.

2 RELATED WORK

Few papers have addressed the problem of semantic scholarly entity extraction. FacetGist [10] is a framework for extracting key facets of a technical document, such as application, technique, evaluation metrics, and dataset. The documents are abstracts from 51,897 DBLP and 11,203 ACL papers. The entities extracted are all noun phrases. Other work by Gupta & Manning[5] extracted three types of information from the abstracts of 474 NLP papers, namely the focus, the domain of application, and the techniques. The extraction is done by semantic pattern matching on the dependency trees of the sentences in an article's abstract. While some of the extracted phrases are labeled as NN (noun compound modifier dependency), some start with gerunds (i.e., V-*ing*) or even prepositions (e.g., for).

While there are many non-sequential keyphrase extractors, we investigate three popular ones to use on scientific documents: RAKE, Maui, and CeKE. RAKE[8] is an unsupervised approach that uses centrality measures on *word collocation graph* to produce a ranked list of keywords. Maui's architecture resembles that of many other supervised keyphrase extraction systems[7]. It first determines textual sequences defined by orthographic boundaries and splits these sequences into tokens. Next, machine learning models are built to calculate probabilities that the candidate is indeed a keyword. CeKE is another supervised machine learning approach. In addition to the existing features for keyphrase extraction, such as tf-idf and POS tags, it uses citation network based features. This makes it non-trivial to implement on our data. We applied RAKE and Maui on a small sample in the training set, and found that they both deliver extremely low recall and precision.

Gollapalli et al. trained a CRF using orthographic and shallow syntactic features [3] to extract keyphrases. Our work is different in two aspects. First, the non-sequential methods in their work is *unsupervised*, where we use *supervised* methods trained on the ground truth data. This makes a significant difference in terms of precision, and the F1-measure increases by 6%. In contrast, their F1-measure[3] increases by only about 1% for one dataset, and even decreases for the other dataset. Second, their ground truth[3] was adopted from Caragea[1] for which keyphrases are assigned by authors. In contrast, our ground truth comprised of *semantic entities* (see Section 3.1 for their differences).

3 DATASET AND EVALUATION PROTOCOL

3.1 Dataset

We use the dataset released in the SemEval2017 competition Task 10 (i.e. "Extracting Keyphrases and Relations from Scientific Publications")[*]. This corpus, collected from ScienceDirect, consists of 500 passages, each of which is in a journal article in Computer Science, Material Science, or Physics. The entire dataset was divided into training, development and testing datasets. Here, we refer to each *passage* as a *document* because labeling and prediction are only performed on passages, although the full text is provided for training and development datasets. There are 350 documents in the training set, 50 in the development set, and 100 in the testing set. Because the gold standard and full text are not released for the testing set, for this work we treat the development set as the testing set. The longest passage in the training set contains 264 words and the shortest contains only 60 words. There are in total 6732 phrases labeled in the training set and 1128 phrases in the testing set.

Consider the sentence:

> We consider the shape optimisation of two- and three-dimensional solids by combining multiresolution subdivision surfaces with immersed finite elements.

the underlined spans are annotated as SDKEs and their character boundaries [(16,34),(38,71),(75,121),(127,151)] with respect to the beginning of the passage are given in the corresponding annotated file[†].

3.2 Evaluation

Our evaluation protocol matches that of the previously mentioned competition. For each paragraph, our system produces an annotated file with the character spans for the phrases we identify as SDKEs. If a predicted phrase span is *exactly* the same as one in the gold standard file, we consider that a match. Note that the evaluation protocol is rather strict. In many cases we extract SDKEs

[*]https://scienceie.github.io/

[†]The gold standard provides phrase types as well, but in this work, we focus on extraction only.

that enclose gold standard phrases, but they are considered as mismatches. To evaluate the systems, precision, recall, and F1-measure are calculated over the entire set of gold standard phrases and the extracted phrases. Precision is defined as $N(\text{match})/N(\text{extracted})$. Recall is defined as $N(\text{match})/N(\text{gold standard})$. F1-measure is defined as the harmonic mean of precision and recall. We train all our models on the training data and report the results on the testing data.

4 KNOWLEDGE EXTRACTION

Our first approach is unsupervised with a high recall but low precision. However, it provides a very strong baseline. Next, we filter the candidate phrases generated by the unsupervised approach by employing a supervised classifier. To exploit the sequential nature of the problem we design a linear chain CRF, which extracts longer phrases with high precision, albeit with a lower recall. Finally, we merge the approaches to improve the F1-measure. We also propose heuristic rules that can improve the classification accuracy.

4.1 EKE: NP-Chunking Based Extractor

Our unsupervised entity knowledge extraction algorithm (EKE) is based on a keyphrase extraction algorithm initially used on scientific papers for ranking domain experts[2]. Given a span of text, the algorithm first tokenizes it with a list of regular expressions. The token list is then tagged by the Stanford POS tagger. Based on the POS taggers, a grammar based chunk parser is applied to separate two types of chunks: (1) nouns and adjectives, terminated with nouns, expressed as

```
NBAR: <NN.*|JJ><NN.*>
```

and (2) two chunks of (1) connected with a preposition or subordinating conjunction, e.g., of/in/by, expressed as

```
NP: <NBAR><IN><NBAR>
```

Typically in traditional keyphrase extraction both original forms (e.g., capitalization, plural forms) and positional information for extracted keyphrases are lost. To maintain our evaluation protocol, we modified EKE to preserve them. Although it is a simple model, EKE outperforms an NP chunker using a MaxEnt model trained on CoNLL data [11]. When tested on our training set, EKE reaches a recall of 48.4% and a precision of 26.08%, while the MaxEnt NP chunker achieves a recall of 20.5% and a precision of 9.87%. Therefore, we use it as our baseline unsupervised method.

4.2 Supervised EKE

EKE achieves a relatively high recall but low precision. Intuitively, the precision can be improved by filtering out unimportant phrases in EKE results. For this, we use a supervised approach where a phrase from EKE is marked true if it appears in the gold standard else false.

Three metrics are typically used to measure the importance of a word in a document: frequency of the word in the document (tf), inverse document frequency or the number of documents the word appears in (idf), and a combination of these measures (tf-idf). PMI score for a phrase gives the probability of the *collocation* of the words, i.e., whether the occurrence of the words in the phrase is random or it satisfies the criteria of non-compositionality, non-substitutability, and non-modifiability. Mathematically, for a bigram (x, y), the PMI score is calculated as $p(x, y)/p(x)p(y)$. The same idea can be extended to n-grams where $n > 2$.

Specifically, for each phrase extracted in the last step, a feature vector is created from the PMI score of the phrase and the tf-idf values of the words composing it. Phrases can be represented as vectors of tf-idf values of their words, but these vectors can have different lengths. To apply a classification algorithm, they must be converted into a fixed length feature vector. We achieve this by creating a vector with the dimension of $3n + 1$ where n is the maximum number of words in a phrase in our data. The first n dimensions store tf values of the words, the second and third n dimensions store the idf and tf-idf values respectively. For phrases with less than n words, this vector is padded with median tf, idf and tf-idf values calculated from the corpus. The last dimension stores the PMI value for the phrase, which is a scalar. A linear kernel SVM is used as the classifier.

4.3 Sequential Labeling of SDKEs with CRF

As shown previously, most non-sequential keyphrase extractors that focus on language constituents such as noun phrases would fail to learn contextual cues. To address this we propose a linear chain conditional random field (CRF).

Each word in the text is annotated as the beginning of an SDKE (B), inside (I) or outside (O) of it. This formulation is known as B-I-O encoding and has been previously used in similar tasks such as named entity recognition and keyphrase extraction [3]. Given an input sequence of words or tokens in a sentence, the goal of the CRF is to learn the sequence of hidden variables, namely, the B-I-O tags and predict the same tags on a previously unseen token sequence. A linear chain CRF models $\Pr(\mathbf{y}|\mathbf{x})$, where $\mathbf{y}$ is a sequence of tags and $\mathbf{x}$ is the input sequence of tokens. K feature functions $(f^1, f^2, \ldots f^K)$ are defined such that each function maps a sequence and a token index to a real number ($f^k(y, x, i) \in \mathbb{R}$.). The global feature vector is defined as $\mathbf{F}(\mathbf{y}, \mathbf{x}) = \sum_i < f^1(y, x, i), \ldots, f^K(y, x, i) >$ [9]. The conditional probability distribution for a sequence pair $(\mathbf{y}, \mathbf{x})$ is given by $\mathbf{Pr}_\lambda(\mathbf{y}|\mathbf{x}) \propto exp(\lambda.\mathbf{F}(\mathbf{y}|\mathbf{x}))$, where λ is the global weight vector. We use a python library[‡] for the implementation of the CRF. The weight vector λ is learned by minimizing likelihood on the training data using a quasi-Newton algorithm (Limited memory BGFS). During inference, the learnt weight vector can produce the probability $\Pr(\mathbf{y}|\mathbf{x})$ for any tag sequence $\mathbf{y}$ on an input sequence $\mathbf{x}$. The most likely sequence (having the maximal probability) is chosen by a dynamic programming approach known as the Viterbi algorithm.

The lexical and syntactic features used for the CRF are motivated by previous work on key phrase extraction [3]. Some of the SKDEs (specifically, *materials*) are chemical name and formulae, therefore, some of our features are borrowed from cheminformatics literature as well. Briefly, the most important features are: 1. The word itself; 2. Words within a boundary of two words, both forward and backward (to utilize the context); 3. Part of speech and chunk tags; 4. The suffix of the word (SKDEs tend to end with suffixes "ion" and "ing" and "ons"). Due to space limitations, we refrain from reporting all features and their motivations. Note that pertaining

[‡]https://pypi.python.org/pypi/sklearn-crfsuite

to the formulation of our CRF the feature functions are calculated for each "token" or word.

4.4 Combining CRF and Supervised EKE

Will a non-sequential classifier (e.g., SVM) and a sequential classifier (e.g., CRF) complement each other for this task? To answer this, we compare precision, recall, and F1-measure of individual approaches against the results of the merged model. Merging is performed by directly combining results from each approach and removing duplicates.

As can be seen from Table 1, supervised EKE improves the precision of unsupervised EKE, while reducing the recall, but the overall the F1-measure increases by a significant margin. As expected, the CRF has the best precision among the extractors. When we combine the CRF results with the supervised EKE results, we can see the recall improves by 16%, indicating that these methods are complementary. To further improve the F1-measure, we propose

Table 1: Comparison of HESKD against baselines.

Approach	Precision	Recall	F1-measure
EKE	26%	54%	35%
Supervised EKE	42%	40%	41%
CRF	46%	33%	39%
CRF + Supervised EKE	39%	56%	46%
HESDK	**46%**	**56%**	**50%**

two heuristic rules. We observe that the CRF has a higher precision, therefore, only the first filtering heuristic is applied to both the CRF *and* the supervised EKE results, the rest are applied just on EKE results. The extractor with the combination of supervised EKE, CRF and these rules are referred to as HESDK (Table 1).

- Punctuation Based Filtering: Presence of punctuation marks such as comma or semicolon at the end or unclosed braces indicates incompletion, therefore, phrases with this pattern are removed.
- One Word Phrase Filtering: A scientific concept phrase ideally should have more than one word. However, certain words such as acronyms, chemical names, formulae are usually important. Therefore, they are identified by regular expressions and *not* removed, even if classified as false by EKE. Also, single words with a tf-idf value above a threshold (determined from the training corpus) are retained.

4.5 Discussion

In many cases, a long SDKE is a combination of phrases extracted by an EKE. For example, in the sentence "From a practical point of view, we have endowed the weighted additive model with a distance function structure...", EKE extracts "weighted additive model" and "distance function structure". In the training set, we have the information whether an EKE phrase is inside, outside or an exact match of a gold standard phrase. We can train a non-sequential classifier to predict the same information on the test data. However, directly using this information will cause over-fitting because on the test data the classifier results will be inaccurate. Therefore, we train the classifier using a probabilistic formulation. As our feature functions are defined for each word in a sequence, if a word in a training sentence is not inside an EKE phrase, the vector is $[0.5, 0.5]$, i.e., drawn from a uniform distribution. Otherwise, the distribution is biased. If it belongs to an EKE phrase that is inside or exact match to a gold standard phrase, it is $[0.5+\epsilon, 1-(0.5+\epsilon)]$ else $[1-(0.5+\epsilon), 0.5+\epsilon]$, where $\epsilon \sim \text{Uniform}(0.5, 1)$. The non-sequential classifier predicts the same probability distribution for EKE phrases extracted from the test data. While this modification improves results significantly in the training set, in the test set the improvements in the CRF results are marginal. We attribute this to the low prediction accuracy of the non-sequential classifier in a multi-class setting, which can be further investigated in the future.

5 CONCLUSION AND FUTURE WORK

Scientific Domain Knowledge Entities are phrases that represent concepts in a scientific document. In this paper, we propose two separate classifiers to extract them from the text of a paper: a non-sequential classifier learning the informativeness of a phrase globally and a sequential classifier learning the same utilizing the local context. We show they can be combined using a hybrid method (HESDK) to improve the accuracy of extraction. Preliminary experiments suggest that informativeness measures can be directly embedded in the sequential model itself; results we hope to further validate.

ACKNOWLEDGMENTS

We gratefully acknowledge partial support from the National Science Foundation.

REFERENCES

[1] Cornelia Caragea, Florin Adrian Bulgarov, Andreea Godea, and Sujatha Das Gollapalli. 2014. Citation-Enhanced Keyphrase Extraction from Research Papers: A Supervised Approach. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL.* 1435–1446.

[2] Hung-Hsuan Chen, Alexander G. Ororbia II, and C. Lee Giles. 2015. ExpertSeer: a Keyphrase Based Expert Recommender for Digital Libraries. *CoRR* abs/1511.02058 (2015). http://arxiv.org/abs/1511.02058

[3] Sujatha Das Gollapalli and Xiao-li Li. 2016. Keyphrase Extraction using Sequential Labeling. *CoRR* abs/1608.00329 (2016). http://arxiv.org/abs/1608.00329

[4] Parthasarathy Gopavarapu, Line C. Pouchard, and Santiago Pujol. 2016. Increasing Datasets Discoverability in an Engineering Data Platform using Keyword Extraction. In *Proceedings of the 16th ACM/IEEE-CS on Joint Conference on Digital Libraries, JCDL 2016, Newark, NJ, USA, June 19 - 23, 2016.* 225–226.

[5] Sonal Gupta and Christopher D. Manning. 2011. Analyzing the Dynamics of Research by Extracting Key Aspects of Scientific Papers. In *Fifth International Joint Conference on Natural Language Processing, IJCNLP 2011.* 1–9.

[6] J.D. Lafferty, A. McCallum, and F. C. N. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data *(ICML 2001).*

[7] 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, EMNLP 2009, 6-7 August 2009, Singapore, A meeting of SIGDAT, a Special Interest Group of the ACL.* 1318–1327.

[8] Stuart Rose, Dave Engel, Nick Cramer, and Wendy Cowley. 2010. *Automatic Keyword Extraction from Individual Documents.* John Wiley & Sons, Ltd, 1–20. DOI:http://dx.doi.org/10.1002/9780470689646.ch1

[9] Dan Roth and Wen-tau Yih. 2005. Integer linear programming inference for conditional random fields. In *Proceedings of the 22nd international conference on Machine learning.* ACM, 736–743.

[10] Tarique Siddiqui, Xiang Ren, Aditya G. Parameswaran, and Jiawei Han. 2016. FacetGist: Collective Extraction of Document Facets in Large Technical Corpora. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, CIKM 2016, Indianapolis, IN, USA, October 24-28, 2016.* 871–880. DOI:http://dx.doi.org/10.1145/2983323.2983828

[11] Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000,* Claire Cardie, Walter Daelemans, Claire Nedellec, and Erik Tjong Kim Sang (Eds.). Lisbon, Portugal, 127–132.