Scientific Paper Dataset Identification

Weiqiu You
CICS
UMass Amherst
wyou@umass.edu

Xinyue Cui
CICS
UMass Amherst
xcui@umass.edu

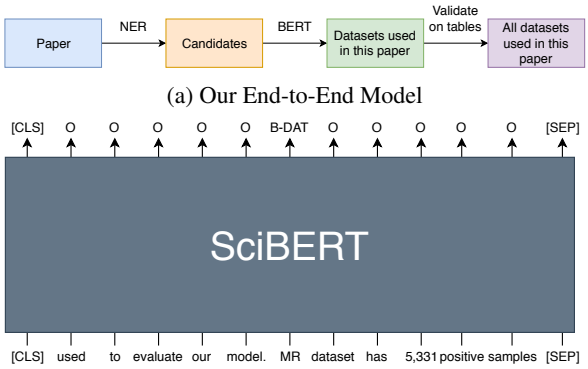
1 Introduction

Most scientific papers in machine learning intend to provide a new methodology for a specific task or problem, which is based and improved on the prior work. While the task, related work, and contribution are usually marked by the phrase "in this paper" in the introduction section, the dataset and metrics are usually described in the experiment section. There is a certain pattern of where we can find these different aspects of a paper. However, finding them by eyeing or simple searches is still very time consuming and not very efficient. We hope to develop some automated methods to identify and categorize the features of the papers, including datasets, metrics, research problems, and so on, which can make researchers easily find the latent related work and help formulate a structural research network.

The prior works try to help people have a quick grasp of a increasing number of publications, thus comparing and mapping the relative papers can be a good idea. We can see previous research involved several aspects including algorithms, concepts, techniques, citations, etc. And the following outcomes are various, such as diagrams like roadmap, categories of concepts. However, the form of output is to some extent restricted and does not apply very well to identifying unseen datasets or research methods in the training set.

This project mainly focus on identifying the datasets, while our method could be used to extract other aspects of the paper in the future. Given a paper, we want to identify not only datasets mentioned in the paper, but also the ones that are actually used in their research.

Our method starts by extracting candidate datasets from papers by named entity recognition. We mainly use rule-based method to extract the candidates using regular expressions. After that,



(a) Our End-to-End Model

(b) Sci-BERT Inputs and Outputs

Figure 1: Model

we finetune on SciBERT to use the context information of the named entities to eliminate ones that are not actually datasets, and keep ones that are real datasets. In the end, if the paper has a LaTeX file provided, we would validate on the tables of LaTeX file and see if we have missed any dataset. If we extracted one dataset but not another but both appear among the same row or column headers, we would add the other one also to our list of datasets.

2 Approach

We use a rule-based baseline as well as our model that combines both rule-candidate extraction and neural-based validation.

2.1 Baseline

We use a simple rule-based model as our baseline. Name of a dataset is usually a capitalized word or acronym. Within those words, by looking at papers in the training data, we see some positional rules that often works:

- A capitalized word in front of the word "dataset" is usually a dataset.

- A capitalize word right after the word "dataset" is usually a dataset.
- When there is a pattern "A and B", if A is a dataset, then B is usually a dataset, vice versa.
- If a few words are connected by commas, and one of them is a dataset, then the rest are usually also datasets.

This covers a lot of the cases we see by looking at the training data.

2.2 Our Model

Our end-to-end model can be shown in Figure 1 (a). The first part of our model is very similar to the rule-based baseline. We extract all words that are capitalized, without distinguishing whether they are near the word "dataset" or another dataset name. We want to get as high recall as possible in this stage.

Examples in candidate extraction would be:

- acronyms: "MR"
- proper nouns: "Movie Review"
- captions of diagram
- headings of diagram

The next step is illustrated in Figure 1 (b). We fine-tune on Sci-BERT to identify whether or not the candidates we extract from the first step are actual datasets. When we train the model, we search for the ground truth dataset names in the paper, and look for five words in front of the words and five words after. This would give us a window size of 11. We put this segment of text as input, and output are labels for each word in the text segment. If the word is the beginning of a dataset, then we output "B-DAT". If the word is in the middle of a dataset name or the end of a dataset name, then we output "I-DAT". For words unrelated to datasets, we output "O". During inference time, we extract window size of 11 around the candidates we extract with rules in the first step. If the candidates are labeled as "B-DAT" or "I-DAT", then we know they are real dataset names. If they are labeled as "O", then we remove them from the candidate list.

The last step is not always used. When we can find the original LaTeX file, we can extract tables from the paper. We have searched for tools to extract tables from PDFs, but none of them work well. Therefore, we decide to only apply this step when we have the LaTeX file. When we have the tables from a paper, we can validate on those tables whether or not we have missed any datasets. If one of the row/column headers is a dataset, then we can say that there is a large probability that the other row/column headers are also datasets. In this way, we can add datasets that we missed to our list. For example, if we have we have "MSB", "AQ", "ACE", "CWEB" and "WW" all as column headers, and we detected with rules and Sci-BERT that "MSB" is a dataset, but missed "AQ", "ACE", "CWEB" and "WW", we can see that they are all column headers along with "MSB" which we have found. Then we will be able to add them all.

	Pairs of dataset-paper	SciBERT Input Lines
Train	4098	532554
Dev	516	33262
Test	509	83037

Table 1: Dataset Split

The data we used is illustrated in Figure 2. (a) is our data in its original PDF form. (b) is how it looks when fed in our model.

3 Experiments

3.1 Datasets

We use the Papers with Code (PapersWithCode, 2019) dataset that is an annotated dataset containing links to papers and listing datasets, metrics and other properties of the papers. This dataset is publicly available online. They have preprocessing code available but not train / dev / tests split. We split the data to be train, dev, and test sets. The proportion is roughly 8:1:1. We first form our data into the form of dataset-paper pairs, and then split them into the three splits, with disjoint papers and disjoint datasets, which means if a paper is in one split, it will not be in any other. It is the same case for datasets.

The number of dataset-paper pairs is in the first column of Table 1. We first split the whole dataset-paper graph into different connected components, and then add connected components to each split from ones with most items to ones with least, until we have the number of items corresponding to the split proportion.

A Multi-sentiment-resource Enhanced Attention Network for Sentiment Classification

Zeyang Lei[1,2], Yujiu Yang[1], Min Yang[3], and Yi Liu[2]
Graduate School at Shenzhen, Tsinghua University[1]
Peking University Shenzhen Institute[2]
Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences[3]
leizy16@mails.tsinghua.edu.cn, yang.yujiu@sz.tsinghua.edu.cn,
min.yang1129@gmail.com, eeyliu@gmail.com

Abstract

Deep learning approaches for sentiment classification do not fully exploit sentiment linguistic knowledge. In this paper, we propose a Multi-sentiment-resource Enhanced Attention Network (MEAN) to alleviate the problem by integrating three kinds of sentiment linguistic knowledge (e.g., sentiment lexicon, negation words, intensity words) into the deep neural network via attention mechanisms. By using various types of sentiment resources, MEAN utilizes sentiment-relevant information from different representation subspaces, which makes it more effective to capture the overall semantics of the sentiment, negation and intensity words for sentiment prediction. The experimental results demonstrate that MEAN has robust superiority over strong competitors.

1 Introduction

Sentiment classification is an important task of natural language processing (NLP), aiming to classify the sentiment polarity of a given text as positive, negative, or more fine-grained classes. It has obtained considerable attention due to its broad applications in natural language processing (Hao et al., 2012; Gui et al., 2017). Most existing studies set up sentiment classifiers using supervised machine learning approaches, such as support vector machine (SVM) (Pang et al., 2002), convolutional neural network (CNN) (Kim, 2014; Bonggun et al., 2017), long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997; Qian et al., 2017), Tree-LSTM (Tai et al., 2015), and attention-based methods (Zhou et al., 2016; Yang et al., 2016; Lin et al., 2017; Du et al., 2017).

Despite the remarkable progress made by the previous work, we argue that sentiment analysis still remains a challenge. Sentiment resources including sentiment lexicon, negation words, intensity words play a crucial role in traditional sentiment classification approaches (Maks and Vossen, 2012; Duyu et al., 2014). Despite its usefulness, to date, the sentiment linguistic knowledge has been underutilized in most recent deep neural network models (e.g., CNNs and LSTMs).

In this work, we propose a Multi-sentiment-resource Enhanced Attention Network (MEAN) for sentence-level sentiment classification to integrate many kinds of sentiment linguistic knowledge into deep neural networks via multi-path attention mechanism. Specifically, we first design a coupled word embedding module to model the word representation from character-level and word-level semantics. This can help to capture the morphological information such as prefixes and suffixes of words. Then, we propose a multi-sentiment-resource attention module to learn more comprehensive and meaningful sentiment-specific sentence representation by using the three types of sentiment resource words as attention sources attending to the context words respectively. In this way, we can attend to different sentiment-relevant information from different representation subspaces implied by different types of sentiment sources and capture the overall semantics of the sentiment, negation and intensity words for sentiment prediction.

The main contributions of this paper are summarized as follows. First, we design a coupled word embedding obtained from character-level embedding and word-level embedding to capture both the character-level morphological information and word-level semantics. Second, we propose a multi-sentiment-resource attention module to learn more comprehensive sentiment-specific sentence representation from multiply subspaces

(a) Our Data - Paper in PDF Form

A Multi-sentiment-resource Enhanced Attention Network for Sentiment Classification Zeyang Lei1,2, Yujiu Yang1, Min Yang3, and Yi Liu2 Graduate School at Shenzhen, Tsinghua University1 Peking University Shenzhen Institute2 Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences3 leizy16@mails.tsinghua.edu.cn, yang.yujiu@sz.tsinghua.edu.cn, min.yang1129@gmail.com, eeyliu@gmail.com Abstract Deep learning approaches for sentiment classification do not fully exploit senti- ment linguistic knowledge. In this paper, we propose a Multi-sentiment-resource Enhanced Attention Network (MEAN) to alleviate the problem by integrating three kinds of sentiment linguistic knowledge (e.g., sentiment lexicon, negation words, intensity words) into the deep neural net- work via attention mechanisms. By us- ing various types of sentiment resources, MEAN utilizes sentiment-relevant infor- mation from different representation sub- spaces, which makes it more effective to capture the overall semantics of the sen- timent, negation and intensity words for sentiment prediction. The experimental results demonstrate that MEAN has robust superiority over strong competitors. 1 Introduction Sentiment classification is an important task of natural language processing (NLP), aiming to classify the sentiment polarity of a given text as positive, negative, or more fine-grained classes. It has obtained considerable attention due to its broad applications in natural language process- ing (Hao et al., 2012; Gui et al., 2017). Most existing studies set up sentiment classifiers using supervised machine learning approaches, such as support vector machine (SVM) (Pang et al., 2002), convolutional neural network (CNN) (Kim, 2014; Bonggun et al., 2017), long short-term mem- ory (LSTM) (Hochreiter and Schmidhuber, 1997; Qian et al., 2017), Tree-LSTM (Tai et al., 2015), and attention-based methods (Zhou et al., 2016; Yang et al., 2016; Lin et al., 2017; Du et al., 2017). Despite the remarkable progress made by the previous work, we argue that sentiment analysis still remains a challenge. Sentiment resources in- cluding sentiment lexicon, negation words, inten- sity words play a crucial role in traditional senti- ment classification approaches (Maks and Vossen,

(b) Extracted Text from PDF

Figure 2: Data

3.2 Baselines

Result of our baseline is in Table 2. We can see that it has a medium recall of 52% but a really low precision of 8%.

Recall	0.52
Precision	0.08
F1	0.14

Table 2: Baseline Performance

When we look at some examples when the model is not able to extract the dataset, we see that there are some different scenarios in Table 5.

Recall	0.96
Precision	0.01
F1	0.02

Table 3: Extract Candidate Performance

Recall	0.89
Precision	0.90
F1	0.89

Table 4: SciBERT Performance

The first example shows how the baseline fails when there is some spelling mistake in the ground truth dataset name. There is nothing much we can do about this. The second example is when the dataset name is the first word in the abstract. The regular expression probably has some problem when extracting phrases from the beginning of the sentence. For the third example, the ground truth dataset is the full name but the one used in the paper is an acronym. In this case, we do extract correctly, but need better way then string matching to match the groundtruth.

3.3 Our Model

The candidate extraction part has a very low precision but high recall (Table 3. This is what we want, because we need to have a high recall for the SciBERT to validate on all the possible candidates.

The SciBERT part has an F1 score of 89%, precision of 90% and recall of 89%. We have not connected this part with the candidate extraction yet.

We can also see from Table 6 one example of where SciBERT failed. The model is supposed to predict "BSD68" as the dataset but it predicts "and". The window contains a lot of proper nouns and the model is probably confused about which one is the dataset. Also, it is even hard for human to know from this small context. Maybe we should increase the window size.

3.4 Software

We use python package tika to extract text from pdfs. For SciBERT, we use Huggingface's implementation of BERT.

Ground Truth Dataset	Failed Reason
IDHP	Extracted: This, IHDP, Todd, A, Analysis, A5, Stefan, ITE, Jobs, A4, The. Failed because of misspelling ground truth
SLAM 2018	Appeared in the first sentence of abstract: SLAM 2018 focuses on predicting a student's mistake while using the Duolingo application. The model could not extract when a dataset is in the beginning of a sentence.
Numenta Anomaly Benchmark	The paper uses the abbreviation "NAB" instead of the full name. Need better matching.

Table 5: Baseline Failed Examples

Text	Predicted	True Label
and	B-LOC	O
FFDNet	O	O
20.	O	O
Kodak24	O	O
(http://r0k.us/graphics/kodak/),	O	O
BSD68	O	B-DAT
53,	O	O
and	O	O
Urban100	O	O
11	O	O
are	O	O

Table 6: SciBERT Failed Examples

4 Related Work

Zha et al. (2019) proposes a method called Cross-sentence Attention NeTwork for cOmparative Relation (CANTOR). They first propose candidate algorithms by rules that can detect abbreviated words, and then use tables in papers as weak supervision to derive directional graphs that map the evolution of algorithms.

Gupta and Manning (2011) extracts focus, technique and domain from scientific papers, by matching semantic extraction patterns, learned using bootstrapping, to the dependency trees of sentences in an article's abstract. Combined with precalculated article-to-community assignment, they study how different sub-fields of the computational linguistics community have changed their foci, methods, etc.

Tsai et al. (2013) proposes an unsupervised bootstrapping algorithm to identify and categorize concepts. They also propose a new clustering algorithm to cluster concepts using citation links between papers

Extracting key phrases and relations from scientific publications(Augenstein et al., 2017) seems important for researchers to find the related articles that they may be interested in, which is also an important part of our research to retrieve the types of the dataset from the corpus. Clustering is a popular method applied in the citation network and there is a systematic comparison of different methods (Van Eck and Waltman, 2017). People also focus on annotation structure and try to link the thoughts by analysing the argumentation structures(Kirschner et al., 2015), developing an annotation tool to represent the arguments in small graph structures. We also consider a friendly design of the visualized results produced by the model.

5 Conclusions and Future Work

We are able to get pretty good result with SciBERT and with extracting candidates. We can confidently say that if we connect the two parts together, it will perform better than the baseline model where we just use our observed rules to extract. SciBERT still has a lot of errors, but we can increase the window size or use the actual sentence instead of a window of text in training and inference.

There are more things to do in the future. The first thing is to connect all parts together. In this project, we did not have time to connect candidate extraction and SciBERT, but we should definitely connect and see how it actually performs.

One other thing after the end-to-end model finishes is to link datasets from all papers together and allow adding new datasets. When we have two papers, we want to know if the datasets they are using are the same ones or not, even if they use slightly different spellings. We want to disambiguating datasets with similar acronyms.

This project focused on extracting datasets from the papers, but the same method can be extend to identify research problems, methods, and metrics etc. It will be harder to use hand-written rules to extract those on those tasks, but with our neural method, it will be feasible to transfer to other tasks.

The final goal will be to construct a paper network based on citation network and authorships, algorithm roadmap and dataset linking, or predict one aspect with others due to the relativity. This system will be able to help people doing research by linking papers they are reading to other papers with same datasets, research problems, methods, etc. and make it easier for people to find papers in research.

References

Augenstein, I., Das, M., Riedel, S., Vikraman, L., and McCallum, A. (2017). Semeval 2017 task 10: Scienceie-extracting keyphrases and relations from scientific publications. *arXiv preprint arXiv:1704.02853*.

Gupta, S. and Manning, C. D. (2011). Analyzing the dynamics of research by extracting key aspects of scientific papers. In *Proceedings of the 5th International Joint Conference on Natural Language Processing*, pages 1–9.

Kirschner, C., Eckle-Kohler, J., and Gurevych, I. (2015). Linking the thoughts: Analysis of argumentation structures in scientific publications. In *Proceedings of the 2nd Workshop on Argumentation Mining*, pages 1–11.

PapersWithCode (2019). Papers with code. https://paperswithcode.com/about. Accessed: 2019-09-26.

Tsai, C.-T., Kundu, G., and Roth, D. (2013). Concept-based analysis of scientific literature. In *Proceedings of the 22nd ACM international conference on Information & Knowledge Management*, pages 1733–1738.

Van Eck, N. J. and Waltman, L. (2017). Citation-based clustering of publications using citnetexplorer and vosviewer. *Scientometrics*, 111(2):1053–1070.

Zha, H., Chen, W., Li, K., and Yan, X. (2019). Mining algorithm roadmap in scientific publications. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1083–1092.