Rich Context Competition: Extracting Research Context and Dataset Usage Information from Scientific Publications

Hyungrok Kim, Kinam Park, and Sae Hyong Park
School of Computing, KAIST, Republic of Korea
{hyungrok.kim, parkinam321, labry}@kaist.ac.kr

ABSTRACT

We describe the task of Rich Context Competition which comprises of extracting research fields, research methods, and the dataset usage information from scientific research papers. It is crucial for researchers and analysts who want to use data for evidence and policy to easily find out who else worked with the data, on what topics and with what results. We tackled this task by dividing it into three different sub-problems. Research field and research method extraction are solved with an unsupervised graph-based classification of publications. Dataset information extraction is done by a sequential scheme of grasping dataset mention phrases from the pure texts, and classifying specific referenced dataset. Although the complexity of the task is high, our approach is relatively compact without much performance decrease.

1 INTRODUCTION

The task from Rich Context Competition [1] was built with the hope to develop the best text analysis and machine learning techniques to discover relationships among datasets, researchers, publications, research methods, and fields. The goal of this competition is to automate the discovery of research datasets and the associated research methods and fields in social science research publications, identify the datasets used in a corpus of social science publications and infer both the scientific methods and fields used in the analysis and the research fields. The final goal is to build a platform with set of tools that enables collaborative knowledge creation and discovery with confidential microdata.

There were three tasks We needed to tackle: 1) Extracting research field for each publication, 2) Extracting research method for each publication, 3) Extracting used datasets and the usages' mention phrases.

For extracting research methods and fields 1), 2), since there were no annotated dataset, we needed to treat it as unsupervised information extraction. For 3) dataset usage info, using the annotated dataset given from [1], it has been a sequential task of key phrase labeling and classification to certain datasets.

Our main objective is to make a simple model without a performance loss. We have accomplished the goal of building a simpler model that achieves high accuracy. According to our analysis, there is an important issue that makes our overall performance decrease and we present the issue in the **Conclusion** section.

2 RELATED WORK

Information extraction from scientific publications has been a wide open task as in SemEval 2017 Task 10 [2]. The main task at [2] is divided into three smaller tasks. From each scientific publications, the tasks are to extract which task the author worked with, which material they used, and which process they used. Except for the material extraction, the participants needed to use unsupervised key phrase extractions to get the process and the task, which was similar to our competition task.

Most of the participants [3][4] combined keyword extraction and labeling model into their approaches. A Well-known neural tagging structure [5] has been the most used key-component to their sequence labeling tasks. For the unsupervised keyword extraction and to put wide-range context to the model, [6] showed a way of using relation network to incorporate all of the task into one multi-tasking model.

On the other hand, several recent studies have tackled the information extraction from documents by generative models. [7] has proposed a neural framework for documents, based on topic models, to enable flexible incorporation of metadata and allow for rapid exploration of alternative models, observing improving performance by the incorporation of metadata. In the other hand, one family of VSL (variational sequential labeler) model [8] for NER (Named Entity Recognition) task has shown a similar performance as [5].

However, our major concern was about getting fast training and inference. Since train data and test data are both 5,000 publications and approximately 1M sentences in such amount of publications each, we needed to make the model as simple as possible to do the training within the given period of time. We needed to make sure the inference for test data should finish within 24 hours to meet the time limitation given from Initiative

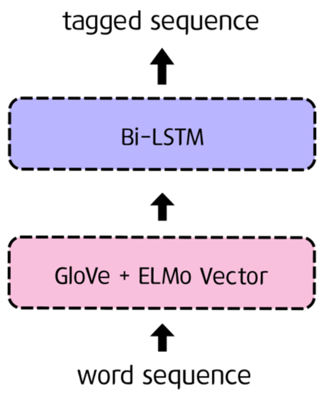


Figure 1: Simple LSTM Labeler with ELMo

With the model shown in Figure 1, in a recent paper [9] has shown that by combining ELMo vector [10] with GloVe [11] and using the concatenated vector as word embedding, the basic Bi-directional LSTM model got the state-of-the-art performance in metaphor detection, which is a type of sequence labeling tasks. The model performed well without a CRF-layer, because ELMo has enough sequential context within the text. We decided to further develop this idea to make a compact and efficient model for the dataset usage info extraction task.

For the task of extracting research fields and research methods, rather than just extracting keywords from the publications, we decided to use a bunch of research field and research method examples (e.g. Public Health, Media Ethics,...;Evidence-based practice, Game theory,...) We use several top extracted keywords to compare the similarities between keywords and those pre-built methods and fields. In this way, we expected to see more consistent results.

3 SOLUTION

Among three tasks, 1) research field extraction, 2) research method extraction, 3) dataset usage information extraction, 1) and 2) will be solved with similar model, and 3) will be taken by another.

3.1 Research Field/Method Extraction

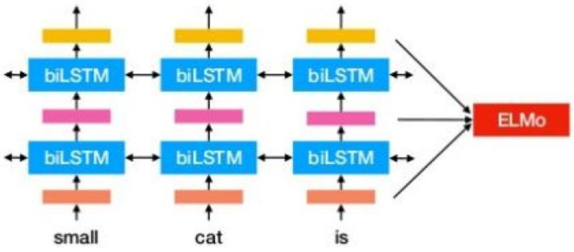
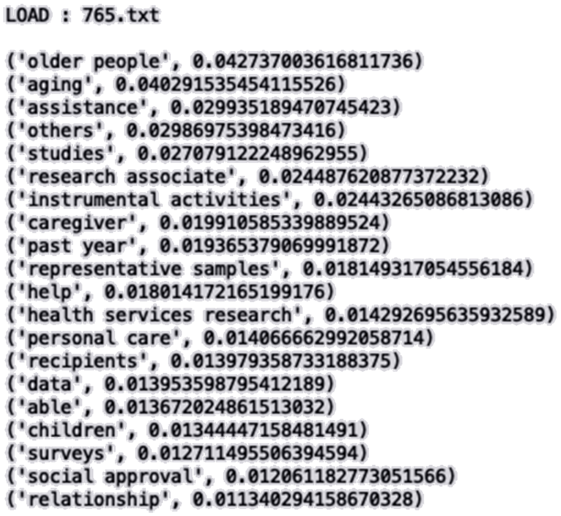


Figure 2: ELMo structure



```
LOAD : 765.txt

('older people', 0.042737003616811736)
('aging', 0.040291535454115526)
('assistance', 0.029935189470745423)
('others', 0.02986975398473416)
('studies', 0.027079122248962955)
('research associate', 0.024487620877372232)
('instrumental activities', 0.02443265086813086)
('caregiver', 0.019910585339889524)
('past year', 0.019365379069991872)
('representative samples', 0.018149317054556184)
('help', 0.018014172165199176)
('health services research', 0.014292695635932589)
('personal care', 0.014066662992058714)
('recipients', 0.013979358733188375)
('data', 0.013953598795412189)
('able', 0.013672024861513032)
('children', 0.01344447158481491)
('surveys', 0.012711495506394594)
('social approval', 0.012061182773051566)
('relationship', 0.011340294158670328)
```

Figure 3: Top 20 KeyPhrase extraction from pke module

3.1.1 Background. **ELMo**[10] is a state-of-the-art contextual word embedder. As Figure2, The BiLM in ELMo uses a deeply stacked RNN structure running in both directions, which can yield context dependent results. In addition, ELMo can catch the meaning of a word using weighted-sum of several hidden layers rather than only using the last layer result. The formula for weighted-sum is like this:

$$ELMO_k^{task} = E(R_k; \theta^{task}) = \gamma^{task} \sum_{j=0}^{L} s_j^{task} h_{k,j}^{LM}$$

Because of ELMo's ability to catch context dependent meanings, we were expecting that the field/method we are looking for would deal well with the case of not being perfect matching. To use ELMo, we adopt the allennlp[1] module to use ELMO. We trained the BiLM model provided by allennlp with 5000 given scientific publications to do its own word embedding.

TopicRank[12] is an unsupervised graph-based key extraction method. TopicRank allows you to graphically organize the association between words to extract keyphases with high weight. The reason for adopting this module is that it can solve the time limit of this competition. If we are scanning all the words and looking for fields and methods, it will take a lot of time. Therefore, key word extraction can dramatically reduce the time required for field/method finding by selecting important words. To use this function, we take pke[2] module which can extract highest weighted topic in decreasing order as Figure3. We applied this module to each document to create a dataset that consisted only of the top 20 keyphrases.

To construct efficient inferring, we have to see **Structure of Sage Research Field and Method file**. These files have a list of field/method which we want to match. The sage-research-field file has a hierarchical structure. Therefore, we can reduce the complexity to log (n) by traversing from the upper-tier field to the lower-tier in order. On the other hand, since the sage researh method is not hierarchically structured, it takes too much time to travel all of them. Therefore, if the occupation rate of a specific method becomes very high in the analogy process, the method travel is stopped and the method is confirmed.

3.1.2 Approach. With both key phrase and sage research field method, we can get the embedded ELMo vector in which dimension is 1024. Using these vectors, we can see how each key phrase has a similar meaning to a particular field. We will take the cosine distance between the two vectors and assume that the field with the shortest distance is the field that represents the publication.

3.2 Dataset Mention Extraction

For the required dataset mention extraction result, we needed to capture all dataset-mentioning phrases from publication's pure text, and also specify which dataset it is mentioning. Our model for the dataset mention extraction is shown in Figure 4. For mention phrase extraction as in the left side, the labeler from [9] extracts the mention-phrase but does not specify which dataset it is. That dataset information gets specified by a basic CNN text classifier [13].

[1] https://github.com/allenai/allennlp/blob/master/tutorials/how_to/elmo.md
[2] https://github.com/boudinfl/pke

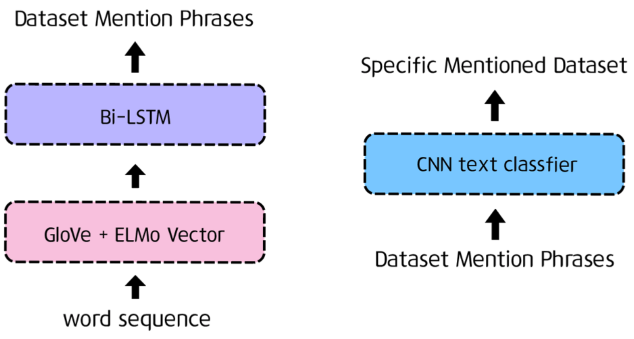


Figure 4: Model Structure for Dataset Mention Extraction

3.2.1 RNN Labeler. As we described, the Bi-LSTM labeler using ELMo+GloVe vector structure is quite simple. For every sentence from publications, the ELMo vector gets extracted, and GloVe word vector gets extracted for every word in the sentence. The concatenated word embeddings becomes the inputs to Bi-directional LSTM. With the output state from LSTM, output linear layer and softmax function computes the probability of each word to be labeled "O", "B", or "I". Words labeled "O" mean the words that are not related to dataset mentions. The words labeled "B" mean the words that are in the beginning position of the mention phrases, and "I" means the words that are inside, or at the end of the dataset mention phrases. Finally, we can extract word sequences that are in form as in Figure 5, "B" and following "I"s.

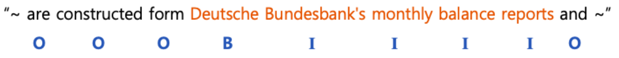

"~ are constructed form Deutsche Bundesbank's monthly balance reports and ~"
O O O B I I I I O

Figure 5: an example of how Dataset Mentions are labeled

3.2.2 CNN Text Classifier. From extracted dataset mention phrases, we used CNN Text Classifier model [13]. First sentences use the same language modeling as before, ELMo+GloVe. The multi-filter structure of CNN model extracts the squeezed states, and gets max-pooled. Then, by the dense linear layer and softmax, the sentences(mention phrases) are computed to probabilities of which dataset it is relating, in our case there were 10,348 different datasets(classes).

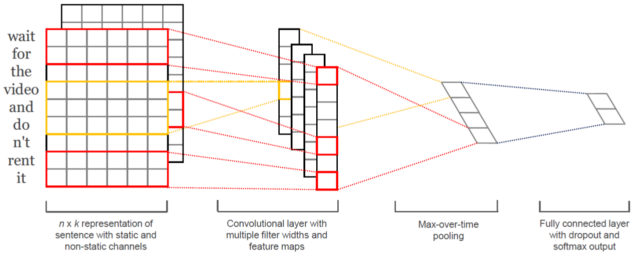

wait
for
the
video
and
do
n't
rent
it
n x k representation of sentence with static and non-static channels
Convolutional layer with multiple filter widths and feature maps
Max-over-time pooling
Fully connected layer with dropout and softmax output

Figure 6: CNN Text Classfier Model Structure

4 EXPERIMENT

7 The experiment progress starts from preprocessing the pure publication texts. Then training the ELMo language model, extracting keyphrase from publication, inferring field method using cosine distance between ebedded ELMo vectors, extracting mention phrases by labeler, and classifying the dataset each mention is relating.

4.1 Preprocessing

4.1.1 Text Normalization. In the given train set and test set size are 5,000 publications each, which is the amount of 3 4M sentences each. First, for consistency, we built our custum rule-based sentence splitter. Then used NLTK word tokenizer to tokenize words. To reduce the size of useless text, we built a bunch of normalization functions using regex, especially ignoring numbers is usual in NLP but we didnâĂŹt just ignore it since the year information means a lot in dataset context, instead we built some pattern rules to get rid of numbers other than years. We ignored words that are longer than 15 characters, and allowed only the sentences with length in 10 âĂŞ 30 words.

4.1.2 Train BiLM in ELMo. To make embedded vector of key phrase and field method, we should train the BiLM in ElMo model. To implement the appropriate embedding on our subject, we trained the BiLM model with a train set of 5000 publications given at the competition. We used three 1080ti GPUs, and training process tooks at least 9 hours. After training, we save the trained weight file and apply it to the ELMo model.

4.1.3 Decrease content considering time complexity. We can infer all the words in the publication, but it is not good for competition because it takes too much time. So we need to shorten the contents of the publication in order to solve it. We expected that the field/method we wanted to extract would necessarily be in the abstract and introduction of publication because it is the keyword. So we decided to extract only 4 50 sentences of each publication and set it as the basic dataset for field/method extraction.

4.2 Inferring field/method

Using pke-TopicRank module, we extract top 20 keyphrases from these dataset. While looping through the top 20 key phrases of each publication, apply ELMo embedding and then calculate the cosine distance between embedded vectors of key phrase and research field to see if there are words of similar meaning. In this process, the research field with the minimum cosine distance is stored and this becomes the research field of each publication. It has a hierarchical structure of the sage research field file, which allows for a rapid traversal of log (n) time complexity. The process of inferring method is almost similar but the way to travel the sage research file is slightly different. Because the method list was not hierarchical, we had to use some tricks. Once a method with a minimum cosine distance is updated, count from that time. Then, if the updated method does not change past 100 iterations, then this method is presumed to be the correct method and is chosen. This process reduced the average time complexity. For 100 publication test set, with three 1080ti GPU, Inferring field method process takes 8 minute. Without GPU, it takes 20 minutes. Also, for 5000 test set corpus to be used in the competition, We can also expect a test time of 5 hours when using the GPU.

Then we save the trained weight file and apply it to the ELMo model. We used three 1080ti GPUs, and training process tooks at least 9 hours.

4.3 Dataset Mention Labeler

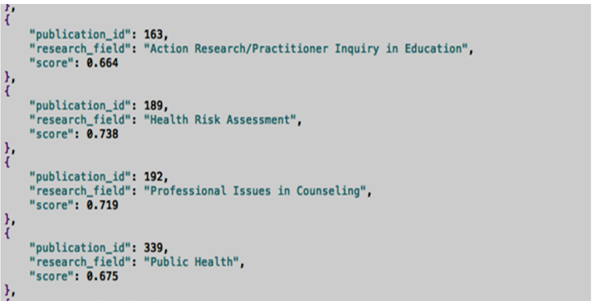
The embedding dimension is 1024 in total, 300 for GloVe, 1024 ELMo and 4 for capital information. Because we are checking mention info, and usually there are lots of upper character usages in dataset names, we decided to include those in the embeddings by one-hot encoding case by case. [1, 0, 0, 0] if all characters are lower case, [0, 1, 0, 0] if all characters are upper case, [0, 0, 1, 0] if only the first character is upper case, [0, 0, 0, 1] otherwise. For the model's configuration, input dropout was 0.5, we used one layer of Bi-directional LSTM with hidden state dimension 300, 0.2 rated dropout on LSTM's output, and one linear layer with LogSoftmax function. For training, we used SGD optimizer + 0.9 momentum + Nesterov, with learning rate decay, and trained for 20 hours with 3 of 1080Ti GPUs. **With the validation set, it reached 99.94% accuracy on word-wise label comparison.**

4.4 Dataset Classifier

For the classifier, we have tried to use Bi-LSTM text classifier also, but stated CNN text classifier worked better. For the model's configuration, the CNN structure had 5 Kernels with sizes 1, 2, 3, 4, 5 (all kernels' second dimention's size is same as the input embedding dimension) with 800 kernels each, and one linear layer, 0.3 dropout, ReLU, Max-pooling, and Softmax function. After the softmax function, we didnâĂŹt just pick from the max probability one, because if the dataset is newer than the paper, it is a conflict, we chased every dataset and publication's published dates and resolved those conflicts by seeing top 10 instances. For training, we used SGD optimizer + 0.9 momentum + Nesterov, with learning rate decay, and trained for 20 hours with 3 of 1080Ti GPUs. **With the validation set, it reached 53.52% accuracy on mention-datset comparison.** Due to lack of time we tested with 100 publications, for overall dataset mention extraction, it took 25 minutes.

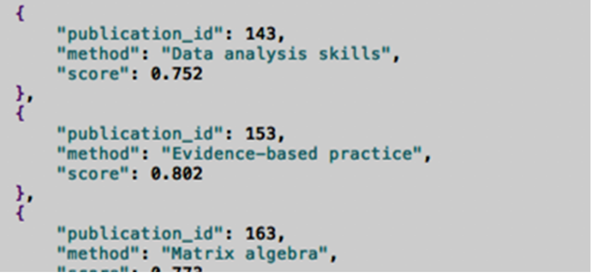
5 RESULT

5.1 Inferring Field and Method



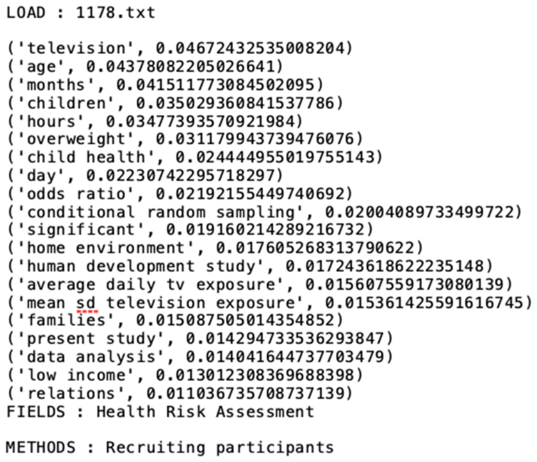
```
},
{
    "publication_id": 163,
    "research_field": "Action Research/Practitioner Inquiry in Education",
    "score": 0.664
},
{
    "publication_id": 189,
    "research_field": "Health Risk Assessment",
    "score": 0.738
},
{
    "publication_id": 192,
    "research_field": "Professional Issues in Counseling",
    "score": 0.719
},
{
    "publication_id": 339,
    "research_field": "Public Health",
    "score": 0.675
},
```

Figure 7: Inferred Research Fields with Scores



```
{
    "publication_id": 143,
    "method": "Data analysis skills",
    "score": 0.752
},
{
    "publication_id": 153,
    "method": "Evidence-based practice",
    "score": 0.802
},
{
    "publication_id": 163,
    "method": "Matrix algebra",
```

Figure 8: Inferred Research Methods with Scores



```
LOAD : 1178.txt

('television', 0.04672432535008204)
('age', 0.04378082205026641)
('months', 0.041511773084502095)
('children', 0.035029360841537786)
('hours', 0.03477393570921984)
('overweight', 0.031179943739476076)
('child health', 0.024444955019755143)
('day', 0.02230742295718297)
('odds ratio', 0.02192155449740692)
('conditional random sampling', 0.02004089733499722)
('significant', 0.019160214289216732)
('home environment', 0.017605268313790622)
('human development study', 0.017243618622235148)
('average daily tv exposure', 0.015607559173080139)
('mean sd television exposure', 0.015361425591616745)
('families', 0.015087505014354852)
('present study', 0.014294733536293847)
('data analysis', 0.014041644737703479)
('low income', 0.013012308369688398)
('relations', 0.011036735708737139)
FIELDS : Health Risk Assessment

METHODS : Recruiting participants
```

Figure 9: Result of inferring fields & methods

Figure7 and Figure8 show the result of inferring fields & method. We were able to match the appropriate field and method according to each publication and gave a score using the cosine distance value. The lower the cosine distance, the higher the score.

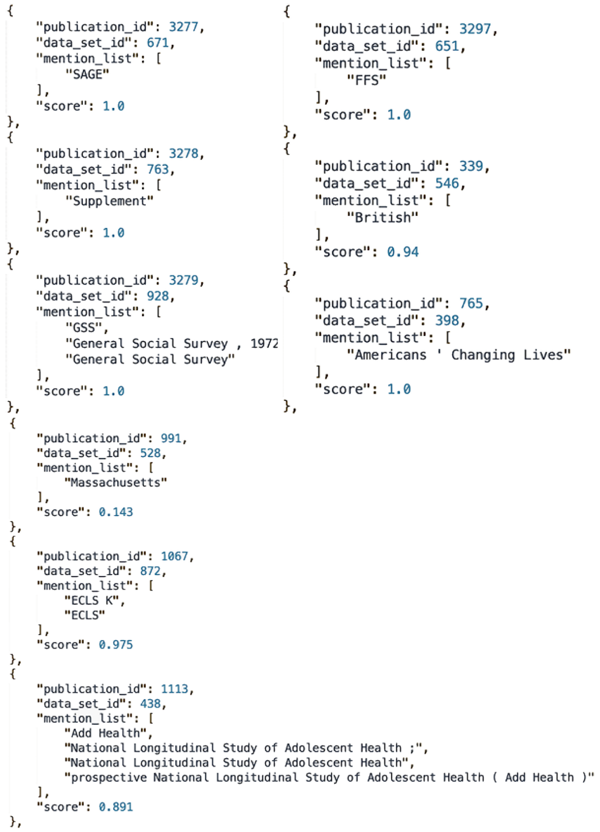
Figure9 shows that the result reflects against 1178.txt file. This paper deals with overweight incidence according to TV watching time and the researchers recruited participants to study the effects of television. We have seen our model estimate the health risk assessment as a field and estimate the recruiting participant as a method, which is a very well-estimated result. Although we did not get high estimates from all the results, we found high accuracy when we obtained meaningful embedded vectors in specific publications.

5.2 Dataset Mention Extraction

Figure 10 shows few samples from our overall result in dataset mention extraction. Before computing recall, precision, and accuracy, it looked good since the most of the mentions and the classified datasets were in match. We checked captured mentions were actually seen in the classified datasets' mentions in history, and those were found to be with very high scores (softmaxed probability).

5.2.1 Evaluation. Then we were able to evaluate with evaluation code given from Initiative. The evaluation algorithm is as followed.

First, we map every mentions that are extracted from ceritain publication, to that publication.



```
{
    "publication_id": 3277,
    "data_set_id": 671,
    "mention_list": [
        "SAGE"
    ],
    "score": 1.0
},
{
    "publication_id": 3278,
    "data_set_id": 763,
    "mention_list": [
        "Supplement"
    ],
    "score": 1.0
},
{
    "publication_id": 3279,
    "data_set_id": 928,
    "mention_list": [
        "GSS",
        "General Social Survey , 1972
        "General Social Survey"
    ],
    "score": 1.0
},
 {
    "publication_id": 991,
    "data_set_id": 528,
    "mention_list": [
        "Massachusetts"
    ],
    "score": 0.143
},
{
    "publication_id": 1067,
    "data_set_id": 872,
    "mention_list": [
        "ECLS K",
        "ECLS"
    ],
    "score": 0.975
},
{
    "publication_id": 1113,
    "data_set_id": 438,
    "mention_list": [
        "Add Health",
        "National Longitudinal Study of Adolescent Health ;",
        "National Longitudinal Study of Adolescent Health",
        "prospective National Longitudinal Study of Adolescent Health ( Add Health )"
    ],
    "score": 0.891
},
```

```
{
    "publication_id": 3297,
    "data_set_id": 651,
    "mention_list": [
        "FFS"
    ],
    "score": 1.0
},
{
    "publication_id": 339,
    "data_set_id": 546,
    "mention_list": [
        "British"
    ],
    "score": 0.94
},
{
    "publication_id": 765,
    "data_set_id": 398,
    "mention_list": [
        "Americans ' Changing Lives"
    ],
    "score": 1.0
},
```

Figure 10: samples from Dataset Mention Extraction Result

For each publication, we will get list of all datasets which are mentioned from that publication.

Compare the predicted dataset list for each publication with the actual dataset list for the publication, if a predicted dataset is not in actual list, it is a False Positive. If a predicted dataset is also in the actual list, it is a True Positive. If an actual dataset is not in the predicted list, it is a False Negative.

	Predicted: Negative	Predicted: Positive
Actual: Negative	-	113 (FP)
Actual: Positive	75 (FN)	25 (TP)

Table 1: Confusion Matrix from Dataset Mention Extraction

Precision	0.18
Recall	0.25
Accuracy	**0.12**

Table 2: Precision, Recall, Accuracy from Dataset Mention Extraction

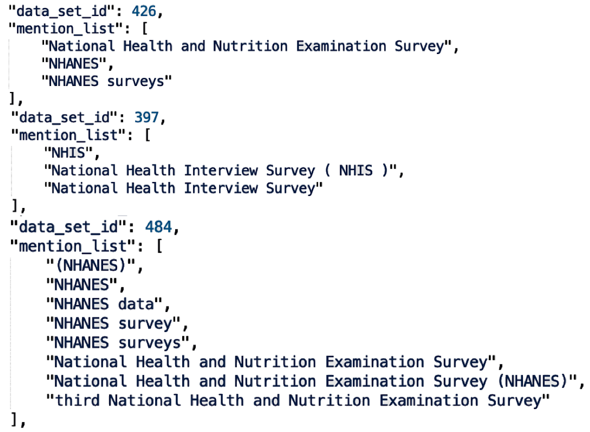
6 CONCLUSION AND FUTURE WORK

6.1 Inferring Field and Method

The Advantage of our approach is that since we used word embedded vector, the model can match key phrase and fields & methods easily. Our model, which is close to exact matching, shows high performance in publication with simple keyword. There is a disadvantage in that the model does not perform very well in the publication in which keyword is described indirectly. However, context-dependent embedding, an advantage of ELMo, guarantees some performance in order to compensate for this. To solve this problem, we need to introduce another technique to analyze the association of each different word, not just depending on the performance of ELMo. The simplest and most effective way is to increase the word pool of synonyms through external crawling.

6.2 Dataset Mention Extraction

In the end, we achieved a very poor accuracy that we never expected.



```
"data_set_id": 426,
"mention_list": [
    "National Health and Nutrition Examination Survey",
    "NHANES",
    "NHANES surveys"
],
"data_set_id": 397,
"mention_list": [
    "NHIS",
    "National Health Interview Survey ( NHIS )",
    "National Health Interview Survey"
],
"data_set_id": 484,
"mention_list": [
    "(NHANES)",
    "NHANES",
    "NHANES data",
    "NHANES survey",
    "NHANES surveys",
    "National Health and Nutrition Examination Survey",
    "National Health and Nutrition Examination Survey (NHANES)",
    "third National Health and Nutrition Examination Survey"
],
```

Figure 11: Analysis on Dataset Mention Extraction Result

6.2.1 Problem: Too Many Versions of Similar Dataset.

We think the biggest reason is because there are a lot of variations in similar datasets. For example, there are **38 different datasets named *NHANES***. If we look through the result, most of the differences are coming from this gap. It finds same mention phrases as the answer, but the dataset specifications are in diverging a lot. In fact, from Figure 11, the dataset id 426, and 397 are also named the same. Those had only the year differences. Even though we cut off impossible dataset match by year specification, there are too many datasets with the same name (with different published date, or version or specific category).

6.2.2 Reason: Loss of Wide Range Context from Documents.

As we decribed, for efficiency, we modeled based on sentence-unit, and we separated labeling and classification process. The problem is that the input for the classifier are only the mention phrase and the year of the mention phrase's original publication. For example, if "NHANES surveys" extracted from publication at 2010, those two are the only context that goes into the classfier, and there're 38 datasets named "NHANES". It is understandable that our model got the low accuracy.

6.2.3 Future Work: Put in Document Context into Classifier.

One of the possible solutions is the one we described in **Related Work**, building relation networks and a complex multitasking model like [6]. But this is not in direction we wanted to work on. Another possible solution is putting the document(publication) context into the classifier with the extracted mention. Such as the recent work by [14], or hierarchical document modeling as in [15].

REFERENCES

[1] Coleridge Initiative. Rich context competition, 2018. https://coleridgeinitiative.org/richcontextcompetition.

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

[3] Yi Luan, Mari Ostendorf, and Hannaneh Hajishirzi. Scientific information extraction with semi-supervised neural tagging. *arXiv preprint arXiv:1708.06075*, 2017.

[4] Animesh Prasad and Min-Yen Kan. Wing-nus at semeval-2017 task 10: Keyphrase extraction and classification as joint sequence labeling. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 973–977, 2017.

[5] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*, 2016.

[6] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. *arXiv preprint arXiv:1808.09602*, 2018.

[7] Dallas Card, Chenhao Tan, and Noah A Smith. Neural models for documents with metadata. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 2031–2040, 2018.

[8] Mingda Chen, Qingming Tang, Karen Livescu, and Kevin Gimpel. Variational sequential labelers for semi-supervised learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 215–226, 2018.

[9] Ge Gao, Eunsol Choi, Yejin Choi, and Luke Zettlemoyer. Neural metaphor detection in context. *arXiv preprint arXiv:1808.09653*, 2018.

[10] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[11] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *In EMNLP*, 2014.

[12] Adrien Bougouin, Florian Boudin, and Béatrice Daille. Topicrank: Graph-based topic ranking for keyphrase extraction. *International Joint Conference on Natural Language Processing (IJCNLP)*, 2013.

[13] Yoon Kim. Convolutional neural networks for sentence classification. *arXiv preprint arXiv:1408.5882*, 2014.

[14] Keet Sugathadasa, Buddhi Ayesha, Nisansa de Silva, Amal Shehan Perera, Vindula Jayawardana, Dimuthu Lakmal, and Madhavi Perera. Legal document retrieval using document vector embeddings and deep learning. *arXiv preprint arXiv:1805.10685*, 2018.

[15] Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489, 2016.

A SOURCE CODE

https://github.com/justin-labry/rcc-09

B INDIVIDUAL CONTRIBUTION

Hyungrok Kim:

- Text Preprocessing (Parsing and Normalization)
- Study on recent papers to plan our research
- Built Labeler & Classifier, training and inference for Dataset Mention Extraction
- Contributed to the overall presentation plan and to the final report

Kinam Park:

- Train ELMo model using given new train corpus from competition.
- Implemented field & method extraction model using my own approach
- Merge the whole process of model and dedicated Dockerfile to make the program run smoothly.
- Contributed to the overall presentation plan and to the final report.

Sae Hyong Park:

- Train Word2vec model using provided corpus as a baseline
- Implement research field using Word2vec model as a baseline
- Configure and provide private repository for the code development
- Participate in the final presentation and report