Leveraging Capsule Routing to Associate Knowledge with Medical Literature Hierarchically

Xin Liu[1,2], **Qingcai Chen**[1,2,*], **Junying Chen**[1,2], **Wenxiu Zhou**[1,2], **Tingyu Liu**[1], **Xinlan Yang**[1] and **Weihua Peng**[3]
[1]Harbin Institute of Technology, Shenzhen, [2]Peng Cheng Laboratory,
[3]Baidu International Technology (Shenzhen) Co., Ltd
{hit.liuxin,junying.chen.cs}@gmail.com, qingcai.chen@hit.edu.cn
wen.xiu.zhou@outlook.com, {liu-ty,xinlan.y}@foxmail.com,
pengweihua@baidu.com

Abstract

Integrating knowledge into text is a promising way to enrich text representation, especially in the medical field. However, undifferentiated knowledge not only confuses the text representation but also imports unexpected noises. In this paper, to alleviate this problem, we propose leveraging **caps**ule **r**outing to associate **k**nowledge with medical **l**iterature **hi**erarchically (called *HiCapsRKL*). Firstly, *HiCapsRKL* extracts two empirically designed text fragments from medical literature and encodes them into fragment representations respectively. Secondly, the capsule routing algorithm is applied to two fragment representations. Through the capsule computing and dynamic routing, each representation is processed into a new representation (denoted as caps-representation), and we integrate the caps-representations as information gain to associate knowledge with medical literature hierarchically. Finally, *HiCapsRKL* are validated on relevance prediction and medical literature retrieval test sets. The experimental results and analyses show that *HiCapsRKL* can more accurately associate knowledge with medical literature than the mainstream methods. In summary, *HiCapsRKL* can efficiently help selecting the most relevant knowledge to the medical literature, which may be an alternative attempt to improve knowledge-based text representation. Source code is released on GitHub [1].

1 Introduction

Knowledge is known as a triple to describe the relationship (r) between head entity (e_h) and tail entity (e_t) with the format of <e_h, r, e_t>. The popular neural models can improve the ability of learning text representation by integrating knowledge, because they usually lack the ability to learn entities and their relationship in the text. However, the medical literature usually contains multiple knowledge, and not all knowledge is beneficial to its subject. The integration of undifferentiated knowledge into the neural models may reduce the accuracy of medical literature representation. For example, in figure 1, it lists three knowledge existed in the medical literature. But, the third knowledge is redundant to the subject of the literature. Therefore, it is an essential step to determine the hierarchical association between knowledge and medical literature before integrating knowledge.

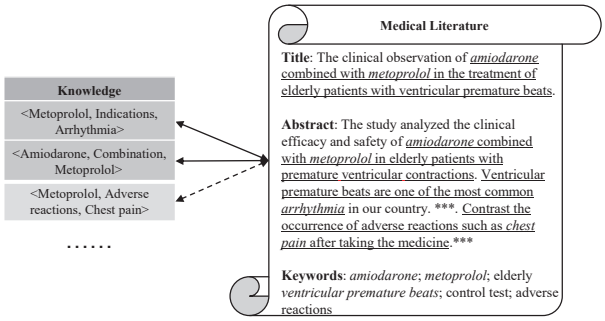

Figure 1: An example to show the undifferentiated knowledge in the medical literature. The literature is from the Chinese Medical Association and is translated from the Chinese version. '***' represents the omitted texts in abstract. The entities are highlighted with italic, and the sentences with underline describe the relationship of two entities.

The hierarchical association between knowledge and medical literature refers to the definition of the degree of their relevance according to how the subject of the medical literature covers the knowledge. Given a knowledge and a medical literature, the task asks to predict their relevance from four levels, namely "*Highly relevance (H_r)*", "*Fairly relevance (F_r)*", "*Marginally relevance (M_r)*", "*Irrelevance (I_r)*" (Kekäläinen, 2005). The public four-point scale graded relevance assessment (Kekäläinen, 2005) from Text REtrieval Conference (TREC) is commonly used for the hierarchical association (See Table 5 in Appendix A.1). But for an intuitive

*Corresponding author.
[1]https://github.com/Gdls/HiCapsRKL

Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, pages 3518–3532
November 7–11, 2021.

Table 1: The definition of *RCor* and *KImp*, and the corresponding relevance labels.

Label	*RCor*	*KImp*
H_r	*positive* (Described)	*Imp* (The knowledge is the only subject the literature discusses.)
F_r	*positive* (Described)	*P-imp* (The knowledge is a subset of the subject the literature discusses.)
M_r	*positive* (Described)	*M-imp* (The knowledge is only mentioned in the literature, and the subject of the literature does not contain more information about it.)
I_r	*negative* (Not described)	*U-imp* (The knowledge is not pointed to the subject of the literature.)

definition, two information measures, namely relationship correlation (*RCor*) and knowledge importance (*KImp*) , should be considered. *RCor* means whether the texts surrounding two entities describe their relationship in the medical literature (Labels: *positive*/*negative*), and *KImp* means how important the knowledge is to the subject of the medical literature (Labels: *Imp*/*P-imp*/*M-imp*/*U-imp*). Table 1 lists the definition of *RCor* and *KImp*, and their corresponding relevance labels. For example, in figure 1, there are always sentences with underlines describing the relationships of "indications" and "combination", and these knowledge is also important to the literature because they are discussed as the subject. But for the third knowledge, even though the *RCor* is positive, it is unimportant to the subject of the literature. So, based on *RCor* and *KImp*, one can easily learn that the top-2 knowledge is highly relevant to the literature and the third one is marginally relevant.

However, it is difficult for the mainstream methods to capture these two information, mainly because 1) the subject of medical literature is usually multi-knowledge entangled, and these methods seldom can learn the unique knowledge from it; 2) the expression of the relationship information in medical literature is complex and abstract, which requires methods with strong distinguishing ability. This paper proposes leveraging the capsule routing algorithm to extract *RCor* and *KImp* information. The capsule routing algorithm is proposed for the capsule network by Sabour et al. (2017), and is an efficient algorithm for decoupling multiple object feature. For the multiple entangled knowledge and complex relationship, the capsule routing algorithm splits the input feature into multiple capsules, and the capsule in a lower-level hands out its output to higher-level capsules through routing algorithm, completing the extraction and aggregation of information flow. After multi-layer capsule calculation, the final layer capsule represents unique knowledge or relation information to solve the issues of knowledge entanglement and complex relationship in medical literature, and then determine the hierarchical association of knowledge and medical literature.

In summary, in this paper our contributions include: 1) proposing hierarchically associating knowledge with medical literature from relationship correlation and knowledge importance, and recording these information through two empirically designed text fragments in the medical literature; 2) proposing leveraging capsule routing algorithm to model the *RCor* and *KImp* text fragments (called *HiCapsRKL*), and taking them as information gain to judge the hierarchical association of knowledge and medical literature; 3) building a weakly supervised training set, a relevance prediction test set and a medical literature retrieval test set, and using these sets to test and analyze the proposed *HiCapsRKL* and other comparison methods. The experimental results and analyses prove the efficiency of *HiCapsRKL* in associating knowledge and medical literature hierarchically.

2 Related Works

The neural information retrieval (IR) models are available techniques for associating knowledge with medical literature because of their powerful deep neural architectures, like CNN (Hu et al., 2014), RNN (Pang et al., 2017), and pre-trained BERT (Devlin et al., 2019). For example, Guo et al. (2016) proposed a joint deep architecture to associate knowledge with medical literature at the query term level. Hui et al. (2017) proposed the position-aware model considering position-dependent interactions. Xiong et al. (2017a) incorporated information from the word space, the entity space, and the cross-space connections through the knowledge. Xiong et al. (2017b) used a translation matrix to model word-level similarities and multi-level soft match features for their association. Dai et al. (2018) used CNN for n-grams of various lengths and soft matched them in a unified embedding

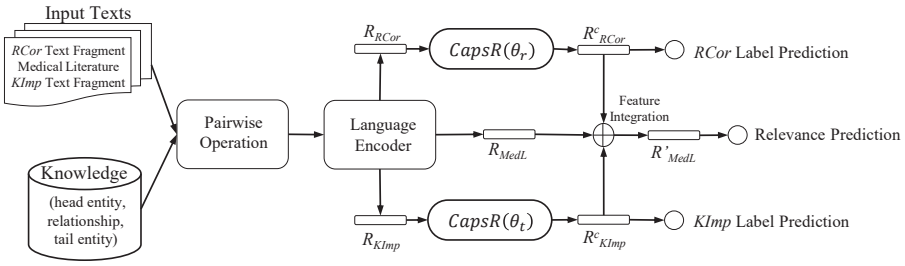

Figure 2: The brief architecture of the proposed *HiCapsRKL* model.

space. Liu et al. (2018b) integrated knowledge to neural models by representing texts with words and entity annotations. Dai and Callan (2019) simulated detecting, determining, and aggregating of human judgement process to associate knowledge and medical literature. MacAvaney et al. (2019a) used the pre-trained contextualized language models to determine their association. However, these works usually focus on learning rich text interaction features based on the end-to-end training, and do not thoroughly explore the association information in knowledge and medical literature (Qu et al., 2019).

For this issue in the former works, the indirect features are used to explore the possible signals in texts. For example, Luo et al. (2017) used the user's click behavior to help the judgement. They believe if there are more user's clicks on one document then the document and knowledge would be more relevant. MacAvaney et al. (2019b) yielded pseudo knowledge-document pairs as relevance indicators that already exhibit relevance. Zheng et al. (2019) followed the heuristics or users' interaction in the result pages to enrich the association features. These methods further improved the neural IR models, but they still did not directly explore the association semantically (Zheng et al., 2018; Zhang et al., 2020).

Relatively, the capsule network is newly proposed neural architecture in recent years and still being explored for its applications in NLP area (Zupon et al., 2020; Nguyen et al., 2019; Zhao et al., 2019). Several researches have explored to apply the capsule network to various NLP tasks, e,g., sentiment classification (Ke et al., 2021; Du et al., 2019b; Chen and Qian, 2019), relation extraction (Liu et al., 2020a), text classification (Chen et al., 2020; Du et al., 2019a; Xiao et al., 2018; Zhao et al., 2018), intent detection (Liu et al., 2019; Zhang et al., 2019; Xia et al., 2018), document translation (Yang et al., 2019), word sense disambiguation (Liu et al., 2020b), etc. Most of these works followed the convolution and dynamic routing architecture in capsule network and did not explore the effectiveness of the capsule routing algorithm for NLP tasks alone (Liu et al., 2020b).

In this work, the proposed *HiCapsRKL* model uses the capsule routing algorithm to learn the relationship correlation and knowledge importance information in texts, and it can semantically explore the hierarchical association of knowledge and medical literature.

3 Method Description

Figure 2 shows the brief architecture of the proposed *HiCapsRKL* model. First, the model inputs include the input texts (i.e. the *RCor* text fragment, the medical literature, and the *KImp* text fragment) and the knowledge triple. Second, each text and knowledge are pair-wised and passed into the language encoder to learn the contextual representation for each pair, namely R_{MedL}, R_{RCor}, and R_{KImp}. Third, two representations R_{RCor} and R_{KImp} go through two capsule routing algorithms, respectively. The algorithm in each branch outputs the corresponding new caps-representation R^c_{RCor} and R^c_{KImp}. Finally, the model integrates three representations for the relevance prediction. Besides, to learn accurate *RCor* and *KImp* features, the model uses each caps-representation to predict its *RCor* or *KImp* label as defined in Table 1 with multi-task training. In this section, the paper will introduce each part of the model in detail.

3.1 Input Pre-processing

First, the input texts include the medical literature, the *RCor* text fragment and the *KImp* text fragment. The medical literature contains the title, abstract and keywords. The *RCor* text fragment is composed of the sentences that simultaneously contain two entities of the knowledge. If two entities do not occur in one sentence, then it is composed of

the sentences that locate between two nearest entities. The *KImp* text fragment is also composed of sentences that contain the title, first sentence in abstract and keywords from the medical literature. By concatenating each sentence in the text, all input text is a sequence of words, namely SEQ_{MedL}, SEQ_{RCor}, and SEQ_{KImp}.

Second, the knowledge triple is input as the concatenation of head entity, relationship and tail entity by using a delimiter, then the knowledge triple is also converted into a sequence of words, namely SEQ_K

Next, the pairwise operation is applied to pair the knowledge triple with the other three input texts, respectively. Then the inputs of the language encoder will be three sequence pairs, namely the medical literature and knowledge pair (<SEQ_{MedL}, SEQ_K>), the *RCor* text and knowledge pair (<SEQ_{RCor}, SEQ_K>), and the *KImp* text and knowledge pair (<SEQ_{KImp}, SEQ_K>).

3.2 Language Encoder

The language encoder in this paper is the pre-trained BERT model initialized with the BERT-Base, Chinese parameters. Three sequence pairs are input into the same BERT model, respectively.

First, in BERT model, each pair is processed with the WordPiece tokenization and sequence concatenation. The first token of every sequence is always a special token ([CLS]), and another special token ([SEP]) is used as the delimiter and end terminator. For example, the input <SEQ_{MedL}, SEQ_K> pair will be converted into the following format <[CLS], token_1^{MedL}, ... , token_m^{MedL}, [SEP], token_1^K, ... , token_n^K, [SEP] > , where token_i^{MedL} and token_i^K means the i-th token in SEQ_{MedL} and SEQ_K. m and n are the maximum token length in each sequence.

Second, the converted sequence goes through the multi-layer Transformer architecture (12 layer in this paper). The model encodes each token with the contextual information and takes the hidden vector in the last layer as the contextual representation for each token. As described in the paper (Devlin et al., 2019), the hidden vector of the special [CLS] token is regarded as the classification representation of the input sequence for down-stream predictions.

Finally, the classification representation of each input sequence for each pair is represented as R_{MedL}, R_{RCor} and R_{KImp}, respectively. The two representations R_{RCor} and R_{KImp} will be processed by the capsule routing algorithm.

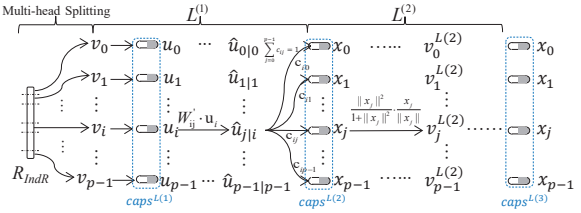

Figure 3: The calculation procedure between initial two layers in capsule routing algorithm.

3.3 Capsule Routing Algorithm ($CapsR(\theta)$)

In this step, R_{RCor} and R_{KImp} are processed with the same capsule routing function but with different initial parameters, namely $CapsR(\theta_r)$ for R_{RCor} and $CapsR(\theta_t)$ for R_{KImp}. Here, this paper only takes the branch of $CapsR(\theta_r)$ as an example to explain the calculation in the function. The calculation procedure between initial two layers is shown in Figure 3.

First, a multi-head splitting operation is applied to the input representation R_{RCor}, and R_{RCor} is split into p sub-vectors with the dimension of D. Multi-head splitting allows cutting the contextual representation into different representation subspaces at different positions (Vaswani et al., 2017). Then R_{RCor} is converted into $\{v_0, v_1, .., v_{p-1}\}$, and each sub-vector v corresponds to one capsule in the first layer. So, for a capsule $caps_i$ in layer $L^{(1)}$ (abbr. $caps_i^{L^{(1)}}$), its input $u_i = v_i$.

Next, a weight matrix W_{ij} with dimensions $D \times D$ is used for building connections with the capsule $caps_j$ in the layer $L^{(2)}$ (abbr. $caps_j^{L^{(2)}}$), and a prediction vector $\hat{u}_{j|i}$ is produced. In $CapsR(\theta_r)$, the parameter θ_t actually refers to the weight matrix W_{ij}. The total input x_j to the capsule $caps_j^{L^{(2)}}$ is a weighted sum over all $\hat{u}_{j|i}$ from the capsules in the layer $L^{(1)}$.

$$x_j = \sum_{i=0}^{p-1} c_{ij} \cdot \hat{u}_{j|i}, \quad \hat{u}_{j|i} = W_{ij} u_i, \qquad (1)$$

where c_{ij} is the coupling coefficient from capsule $caps_i^{L^{(1)}}$ to $caps_j^{L^{(2)}}$. The coupling coefficients sum to 1 between $caps_i^{L^{(1)}}$ and all capsules in $L^{(2)}$, namely $\sum_{j=0}^{p-1} c_{ij} = 1$.

In capsule $caps_j^{L^{(2)}}$, a non-linear "squashing" function as shown in Equation 2 is applied to keep the length by shrinking short vectors to almost 0

and long vectors to a length slightly below 1.

$$v_j^{L^{(2)}} = \frac{||x_j||^2}{1+||x_j||^2} \cdot \frac{x_j}{||x_j||}, \quad (2)$$

where $v_j^{L^{(2)}}$ is the squashing output of the capsule $caps_j^{L^{(2)}}$.

The coupling coefficient c_{ij} is updated by the iterative dynamic routing, and it is a softmax result based on the logic b_{ij}.

$$c_{ij} = \frac{exp(b_{ij})}{\sum_{k=0}^{p-1} exp(b_{ik})}, \quad (3)$$

we follow the processing by Sabour et al. (2017). Initially, b_{ij} equals to 0 and is updated as

$$b_{ij} = b_{ij} + \hat{u}_{j|i}^T \cdot v_j^{L^{(2)}}, \quad (4)$$

which aims to measure the agreement between the output $v_j^{L^{(2)}}$ of $caps_j^{L^{(2)}}$ and the prediction $\hat{u}_{j|i}$ of $caps_i^{L^{(1)}}$.

In the following layers, the function repeats the same calculation. The output $v^{L^{(2)}}$ is passed into the capsules in the next layer and goes through the weight matrix, the weighted sum and the non-linear squashing function. With K layer iterations, we take the outputs of layer K as the relation or topic capsules $\{v_0^{L^{(K)}}, v_1^{L^{(K)}}, ..., v_{p-1}^{L^{(K)}}\}$. Finally, the capsules are concatenated into the new caps-representation R^c_{RCor} for *RCor*. Through the $CapsR(\theta_t)$ function, we have another caps-representation R^c_{KImp} for *KImp*.

3.4 Multi-task Training

After obtaining these representations R_{MedL}, R^c_{RCor} and R^c_{KImp}, the model integrates three representations into the overall representation $R^{'}_{MedL}$ for the relevance prediction. The prediction asks the model to predict the relevance label from "H_r", "F_r", "M_r" and "I_r", and the training loss is marked as $\mathcal{L}_{MedL}$ according to the golden label. To make R^c_{RCor} and R^c_{KImp} learn accurate *RCor* and *KImp* information, the model additionally trains each representation with two specific tasks, namely the *RCor* prediction and *KImp* prediction.

In *RCor* prediction, the model uses R^c_{RCor} as the input feature and predicts the *RCor* label from the binary labels. The binary labels correspond to two cases of the RCor definition in Table 1, namely whether the medical literature describe the relationship of two entities. Therefore, the training loss of this task is marked as $\mathcal{L}_{RCor}$.

Table 2: Comparison in terms of Macro-F1 and Micro-F1 scores (%) considering the label prediction on the relevance prediction test set.

Type	Method	Micro-F1	Macro-F1
Baseline	1.*RCor&KImp*	33.1	24.8
unIR	2.TF · IDF	35.5	30.4
	3.BM25	38.7	33.2
NeuL2R	4.KNRM	40.1	35.2
	5.Conv-KNRM	45.2	40.4
	6.BERT	59.7	49.2
	7.SiameseBERT	59.1	48.4
KGemb	8.MedL+transH	58.3	46.9
	9.MedL+rotatE	58.8	47.9
	10.MedL+transD	59.7	48.6
	11.MedL+transE	58.4	49.2
Ours	12.*HiCapsRKL*	**64.9**	**54.8**

In *KImp* prediction, the model uses R^c_{KImp} as the input feature and predicts the *KImp* label. The *KImp* labels correspond to four cases of the KImp definition in Table 1, namely how important the knowledge is to the subject of the medical literature. Therefore, the training loss of this task is marked as $\mathcal{L}_{KImp}$.

Finally, the total training loss of the model is the sum of three prediction loss, namely $\mathcal{L} = \mathcal{L}_{MedL}+\mathcal{L}_{RCor}+\mathcal{L}_{KImp}$. According to the loss $\mathcal{L}$, the model fine-tunes the parameters in BERT encoder and updates the parameters in capsule routing algorithms during the training.

4 Experiments and Results

4.1 Experimental Datasets and Metrics

In this work, the medical literature is collected from the Chinese Medical Association in 2019, and each literature is represented with a title, an abstract, and keywords. The knowledge triples are from the Chinese medical knowledge graph (CMeKG (Odmaa et al., 2019)).

In the experiment, the training data are automatically constructed based on the *RCor* and *KImp* labels. We first calculated the *RCor* and *KImp* labels between knowledge and medical literature respectively, and then mapped to the relevance labels according to Table 1. Two manually-labeled test sets are proposed to evaluate the *HiCapsRKL* and comparison methods. The knowledge and medical literature in both sets are independent of the training set without any intersections. Both test sets are labeled with professional annotators according to the TREC graded relevance assessment. In the

Table 3: Comparison in terms of P@10, NDCG@10, MRR, and MAP scores (%) on the medical literature test set. The methods in each type are ranked according to NDCG@10. In MRR, MAP, and P@10 metrics, the scores before "/" are calculated based on the "H_r" literature, and the scores after "/" are based on the "H_r" and "F_r" literature simultaneously.

Type	Method	MRR	MAP	P@10	NDCG@10
Baseline	1.*RCor&KImp*	54.6/73.5	43.4/67.7	47.8/67.7	76.1
unIR	2.BM25	53.6/74.8	43.0/70.7	42.6/69.4	74.4
	3.TF · IDF	54.8/76.9	43.3/70.7	43.4/69.9	75.8
NeuL2R	4.KNRM	53.9/78.1	43.2/72.1	42.7/70.2	77.1
	5.Conv-KNRM	54.1/78.9	43.3/72.5	43.5/70.9	77.5
	6.BERT	54.0/83.4	42.4/73.6	39.1/70.8	78.2
	7.Siamese BERT	55.2/83.3	44.6/73.7	42.9/72.2	80.1
KG emb	8.MedL+rotatE	52.7/78.7	43.5/73.1	41.7/71.3	76.7
	9.MedL+transH	57.6/80.3	44.7/73.4	43.2/72.0	78.4
	10.MedL+transD	56.9/82.6	44.7/73.2	43.9/71.3	78.5
	11.MedL+transE	54.7/81.6	44.9/74.5	44.6/73.7	79.2
Ours	12.*HiCapsRKL*	**59.8/83.7**	**46.0/75.2**	**46.6/74.1**	**81.2**

relevance prediction test set, the set asks the model to predict the relevance label of the pair, and the Macro-F1 and Micro-F1 are used as the evaluation metrics. In the medical literature retrieval test set, given a knowledge, the set asks the model to rank the candidate documents based on their relevance to the knowledge. The evaluation metrics on this test set are normalized discounted cumulative gain at 10 (NDCG@10), precision at 10 (P@10), mean reciprocal rank (MRR), and mean average precision (MAP). More details about the data sets are listed in the Appendix A.

4.2 Baseline and Comparison Methods

In this work, the baseline and comparison methods are the *RCor&KImp* baseline, the unsupervised IR methods (unIR, namely TF · IDF and BM25 (Robertson and Zaragoza, 2009)), the neural learning-to-rank models (NeuL2R, namely KNRM (Xiong et al., 2017b), Conv-KNRM (Dai et al., 2018), BERT (Devlin et al., 2019), and Siamese BERT (Reimers and Gurevych, 2019)), and the Translation based KG embedding methods (KGemb, namely transE (Bordes et al., 2013), transH (Wang et al., 2014), transD (Ji et al., 2015), and rotatE (Sun et al., 2019)). More details about the method selection, descriptions, implementations and settings are listed in the Appendix A.

4.3 Experimental Results

Experimental results of all comparison methods on the relevance prediction test set and medical literature retrieval test set are listed in Table 2 and 3.

In Table 2, the *HiCapsRKL* model outperforms all the other methods with Micro-F1 of 64.9% and Macro-F1 of 54.8%. The Micro-F1 indicates that *HiCapsRKL* gives more correct label predictions than other methods, and the Macro-F1 indicates that *HiCapsRKL* brings the overall improvements on four categories because it is the average of the F1 score of each category. Since the limitation of generalization of *RCor&KImp* baseline, it may perform well on the covered cases but perform poor on the un-covered ones. This may be the reason for its poor performance on both metrics. The unsupervised TF · IDF and BM25 methods show similar performance on both metrics, which indicates the upper bound of the unsupervised methods. In NeuL2R methods, CNKM and Conv-CNKM outperform the unsupervised methods with about 5-7% improvement, which mainly benefits from the training set. When the BERT-based models (Line 6-7) are trained, much higher performance is obtained. The knowledge graph embeddings (Line 8-11) contribute a lot to the performance improvement, but they are unstable with about 2% differences compared to BERT. The *HiCapsRKL* model integrates the *RCor* and *KImp* information by capsule routing algorithm, and it reaches the best performance.

In Table 3, the *HiCapsRKL* model also obtains the best performance on all evaluation metrics. The P@10 and NDCG@10 metrics reflect the relevance situation of the top-10 literature among all the retrieved literature, while the MRR and MAP metrics reflect the situation among all the retrieved literature. The NDCG@10 metric not only considers the relevance label of each literature but also considers its position in the top-10 literature. Therefore, it is a

Table 4: The performance in ablation study on both test sets when removing one component from the *HiCapsRKL* model.

Method	Micro-F1	Macro-F1	MRR	MAP	P@10	NDCG@10
HiCapsRKL	64.9	54.8	59.8/83.7	46.0/75.2	46.6/74.1	81.2
w/ CapsR$_{MedL}$	64.1	53.8	59.5/83.4	45.6/74.6	45.1/72.3	80.4
w/o CapsR$_{KImp}$	63.8	53.1	58.8/81.7	45.3/73.6	43.2/72.0	80.1
w/o CapsR$_{RCor}$	63.5	52.5	57.9/82.7	45.4/74.4	44.2/71.7	79.8
w/o CapsR	63.1	50.7	53.3/81.3	42.8/74.1	40.6/71.4	79.1
w/o KImp	62.2	50.1	55.6/81.2	44.8/74.1	44.0/71.3	78.9
w/o RCor	60.5	49.2	54.0/81.4	42.4/73.6	39.1/70.8	78.2
w/o RCor&KImp	59.7	49.2	53.6/80.2	43.3/72.4	42.8/70.4	77.4

comprehensive metric to express the capabilities of the model, and here we use it as the main basis for ranking comparison methods. In the NDCG@10 column, each method can bring a certain improvement. Especially, the *NeuL2R* and *KGemb* methods outperform the unsupervised methods, and benefiting from different mechanisms, they show different improvements. Of all these results, the result of *HiCapsRKL* indicates ranking more literature with higher relevance in the front, and the improvement is a large margin compared to other methods. The P@10 metric counts the literature with a given label in the top-10 literature, and it mainly indicates how much literature that meets the label can be retrieved. Also, from the P@10 column, either on the "H_r" label or on both "H_r" and "F_r" labels, *HiCapsRKL* retrieves the most literature than others. The MAP and MRR metrics report the ranking performance of each method on all the retrieved literature. They mainly report the position of the relevance literature in all literature. The results on these two metrics are roughly consistent to those on the P@10 and NDCG@10 metrics. The results on this set indicate that *HiCapsRKL* is effective for retrieving the relevant literature, and it is also proven to be useful for such tasks and is worthy of further research.

4.4 Significance Test

The significance test was performed based on the comparison methods that were implemented in this paper. The well-known Wilcoxon signed-rank test was used to measure whether the improvement between the corresponding data distributions in two samples are significant. In the Wilcoxon signed-rank test, we first randomly sampled 50% data in each test set for 20 times and used these trained methods to predict the results on the sample data. Second, we scored the sample data with the evaluation script to obtain each metric score. After sampling 20 times, we had a sequence of metric scores with the length of 20 for each method. Finally, the corresponding metric score sequences of any two methods were input into the "wilcox.test()" function in R Tutorial, and the function will output the P-value of two sequences to indicate the significance. If P-value<0.05, the improvement between two methods are significant, otherwise not. Finally, in Table 2 and 3, on both test sets, the improvement between *HiCapsRKL* and any comparison method on each metric is significant (P-value<0.05).

5 Discussion

5.1 Cohen's Kappa Coefficient

Cohen's kappa coefficient (Artstein and Poesio, 2008) is a statistic to measure inter-rater reliability for qualitative items between two categorical variables (McHugh, 2012). In this experiment, we used the coefficient to measure the agreement between the weakly supervised training set and the golden standard.

First, we randomly sampled 25 pairs for each relevance label from the training set, and obtained a random subset with 100 pairs. Second, we manually annotated these pairs. Finally, on the subset, we calculated the Cohen's kappa coefficient score between the automatic labels and the annotated labels. The calculation is completed by the "cohen_kappa_score" function in sklearn toolkit.

The final coefficient score for the random subset is 0.707. Based on the interpretation of Kappa coefficient in Han (2020), the Kappa coefficient score ranging between 0.61 and 0.80 means two variables are "Substantial agreement". The higher the score is, the more perfect the agreement is. For example, the scores ranging between 0.81 and 0.99 means "Near-perfect agreement". The Cohen's kappa coefficient experiment indicates the good quality of the training set. Since the training set is constructed

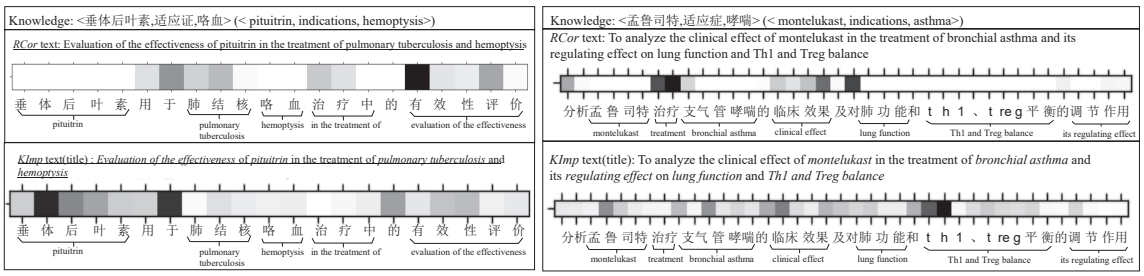

Figure 4: Visualization of two examples for the relationship "indications" to show the attentive weights between each caps-representation and its input text. Their golden labels are "H_r" and "F_r". The cube color in the heat map is darker if the information rely more on these words or characters. For clarity, only the title is listed in *KImp* text.

from a large-scale knowledge and medical literature pairs, it only keep the pairs with high confidence. As a result, the training set presents a higher Kappa coefficient to indicate the substantial agreement with the golden standard.

5.2 Ablation Study

We conducted experiments on removing one component from *HiCapsRKL* to validate how it performs on two test sets. The experimental results are listed in Table 4. The removed component each time is the capsule routing algorithm for *RCor* (CapsR$_{RCor}$), the capsule routing algorithm for *KImp* (CapsR$_{KImp}$), the capsule routing algorithm for both (CapsR), the *RCor* part (RCor), the *KImp* part (KImp), and the *RCor&KImp* parts (RCor&KImp). Especially, an additional experiment of applying capsule routing to R_{MedL} is also included as w/ CapsR$_{MedL}$.

In table 4, from the last 3 lines, we can see that the *RCor&KImp* information plays an important role, and the *RCor* information shows greater influence than the *KImp* information. This is mainly because the relation information is hard to capture in the long medical literature. Moreover, the capsule routing algorithms further improve the performance when they are used for information extraction ("w/o Caps" 3 lines), which indicates that the powerful ability of the capsule routing algorithms. However, it is inappropriate to apply the capsule routing to R_{MedL} ("w/ CapsR$_{MedL}$" line). This is mainly because R_{MedL} is learnt from the entire medical literature, and it is usually asked to learn comprehensive features to determine the hierarchical association, so there is no clear specific feature to extract from it for this task. Overall, all these components are still important in *HiCapsRKL* and contribute to associating knowledge with medical literature.

5.3 *RCor&KImp* Information Visualization

To clearly present how the learned caps-representations in *HiCapsRKL* are related to their input text fragment, we visualized the attentive weights between the caps-representation and its input text fragment. This analysis is performed on the relevance predication test test.

First, we output the caps-representation R^c_{RCor} and R^c_{KImp} in Section 3.3 using the trained *HiCapsRKL* model. Second, we output each token representation in the *RCor* and *KImp* text fragments. Each token representation is from the language encoder. Finally, for *RCor*, we compute the cosine similarities as the attentive weights between R^c_{RCor} and each token representation in *RCor* text fragment, and visualize the attentive weights with heat map. For *KImp*, the computation is between R^c_{KImp} and each token representation in *KImp* text fragment.

Figure 4 lists two typical examples for the most common relationship "indications" due to page limitations, which are randomly selected from all samples to better present the relation between the caps-representation and its input text fragments. In Figure 4, in each example, the heat map in the second block is used for the *RCor* text fragment (*RCor* block), and the third block is for the *KImp* text fragment (*KImp* block). Each block includes the English text translated from its Chinese version, heat map, corresponding Chinese characters and English words. For the relationship, we can see that the characters or words "in the treatment of" in the heat map usually have a larger attentive weight value than others. This indicates that the R^c_{RCor} indeed contains the relationship information. For the subject, the heat map in *KImp* block is absolutely different. In the left example, the subject in the medical literature is about "pituitrin",

"pulmonary tuberculosis", "evaluation of the effectiveness". The knowledge is related to these words, and the words also present larger attentive weight values. The right example also proves the subject information. The heat map of attentive weights indicates that the caps-representations from *HiCapsRKL* have learnt the *RCor&KImp* information in knowledge and medical literature.

6 Conclusion

In this paper, we proposed *HiCapsRKL* to leverage capsule routing to associate knowledge with medical literature hierarchically. This is a worthy research work for better integrating knowledge to learn rich text representation. On two manually labeled test sets, namely the relevance prediction test set and medical literature retrieval test set, the proposed *HiCapsRKL* model has shown SOTA performances than other comparison methods. Exhaustive experimental results and analyses have proven the excellent ability of the proposed model, and showed its potential on learning association features.

In the future, we will focus on applying this work to improve the text representation of the knowledge integration methods by the hierarchical knowledge. For example, the *HiCapsRKL* can be used as multi-task, using the relevance of the knowledge, e.g. the softmax probability or relevance label, as a weight or filter to control the integrating process. *HiCapsRKL* will help to reduce the effect of the noisy knowledge and may further improve the quality of text representation. Besides, this work can also contribute to other NLP researches (e.g., the medical information processing, question answering, information retrieval, reading comprehension, etc), which may benefit from integrating knowledge.

Acknowledgements

We thank all the reviewers for their efforts to make the paper comprehensive and solid. This work is supported by the fund of the joint project with Beijing Baidu Netcom Science Technology Co., Ltd, National Natural Science Foundation of China (Grant No. 61872113, 61876052, 62006061), Special Foundation for Technology Research Program of Guangdong Province (Grant No. 2015B010131010), Strategic Emerging Industry Development Special Funds of Shenzhen (Grant No. JCYJ20180306172232154) and CCF-Baidu Open Fund (Grant No. CCF-BAIDUOF2020004).

References

Ron Artstein and Massimo Poesio. 2008. Inter-coder agreement for computational linguistics. *Computational Linguistics*, 34:555–596.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pages 2787–2795.

Lila Boualili, Jose G Moreno, and Mohand Boughanem. 2020. Markedbert: Integrating traditional ir cues in pre-trained language models for passage retrieval. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1977–1980.

Boli Chen, Xin Huang, Lin Xiao, and Liping Jing. 2020. Hyperbolic capsule networks for multi-label classification. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3115–3124. Association for Computational Linguistics.

Zhuang Chen and Tieyun Qian. 2019. Transfer capsule network for aspect level sentiment classification. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 547–556. Association for Computational Linguistics.

Zhuyun Dai and Jamie Callan. 2019. Deeper text understanding for ir with contextual neural language modeling. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 985–988.

Zhuyun Dai, Chenyan Xiong, Jamie Callan, and Zhiyuan Liu. 2018. Convolutional neural networks for soft matching n grams in ad-hoc search. In *Proceedings of the 11th ACM international conference on web search and data mining*, pages 126–134.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, Jianxin Liao, Chun Wang, and Bing Ma. 2019a. Investigating capsule network and semantic feature on hyperplanes for text classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 456–465. Association for Computational Linguistics.

Chunning Du, Haifeng Sun, Jingyu Wang, Qi Qi, Jianxin Liao, Tong Xu, and Ming Liu. 2019b. Capsule network with interactive attention for aspect-level sentiment classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5489–5498. Association for Computational Linguistics.

Aditya Grover and Jure Leskovec. 2016. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

Jiafeng Guo, Yixing Fan, Qingyao Ai, and W Bruce Croft. 2016. A deep relevance matching model for ad-hoc retrieval. In *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*, pages 55–64.

Xiaoxia Han. 2020. On statistical measures for data quality evaluation. *Journal of Geographic Information System*, 12:178–1872.

Xu Han, Shulin Cao, Xin Lv, Yankai Lin, Zhiyuan Liu, Maosong Sun, and Juanzi Li. 2018. Openke: An open toolkit for knowledge embedding. In *Proceedings of the 2018 conference on empirical methods in natural language processing: system demonstrations*, pages 139–144.

Sebastian Hofstätter. 2020. End-to-end contextualized document indexing and retrieval with neural networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2481–2481.

Baotian Hu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2014. Convolutional neural network architectures for matching natural language sentences. In *Advances in neural information processing systems*, pages 2042–2050.

Kai Hui, Andrew Yates, Klaus Berberich, and Gerard de Melo. 2017. Pacrr: A position-aware neural ir model for relevance matching. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1049–1058.

Guoliang Ji, Shizhu He, Liheng Xu, Kang Liu, and Jun Zhao. 2015. Knowledge graph embedding via dynamic mapping matrix. In *Proceedings of the 53rd annual meeting of the association for computational linguistics and the 7th international joint conference on natural language processing (volume 1: Long papers)*, pages 687–696.

Zixuan Ke, Hu Xu, and Bing Liu. 2021. Adapting BERT for continual learning of a sequence of aspect sentiment classification tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4746–4755. Association for Computational Linguistics.

Jaana Kekäläinen. 2005. Binary and graded relevance in ir evaluations—comparison of the effects on ranking of ir systems. *Information processing & management*, 41(5):1019–1033.

Han Liu, Xiaotong Zhang, Lu Fan, Xuandi Fu, Qimai Li, Xiao-Ming Wu, and Albert Y.S. Lam. 2019. Reconstructing capsule networks for zero-shot intent classification. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 4799–4809. Association for Computational Linguistics.

Tianyi Liu, Xiangyu Lin, Weijia Jia, Mingliang Zhou, and Wei Zhao. 2020a. Regularized attentive capsule network for overlapped relation extraction. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6388–6398. International Committee on Computational Linguistics.

Xin Liu, Qingcai Chen, Chong Deng, Huajun Zeng, Jing Chen, Dongfang Li, and Buzhou Tang. 2018a. Lcqmc: A large-scale chinese question matching corpus. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1952–1962.

Xin Liu, Qingcai Chen, Yan Liu, Joanna Siebert, Baotian Hu, Xiangping Wu, and Buzhou Tang. 2020b. Decomposing word embedding with the capsule network. *Knowledge-Based Systems*, 212:106611.

Zhenghao Liu, Chenyan Xiong, Maosong Sun, and Zhiyuan Liu. 2018b. Entity-duet neural ranking: Understanding the role of knowledge graph semantics in neural information retrieval. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2395–2405.

Cheng Luo, Yukun Zheng, Jiaxin Mao, Yiqun Liu, Min Zhang, and Shaoping Ma. 2017. Training deep ranking model with weak relevance labels. In *Australasian Database Conference*, pages 205–216. Springer.

Sean MacAvaney, Andrew Yates, Arman Cohan, and Nazli Goharian. 2019a. Cedr: Contextualized embeddings for document ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 1101–1104.

Sean MacAvaney, Andrew Yates, Kai Hui, and Ophir Frieder. 2019b. Content-based weak supervision for ad-hoc re-ranking. In *Proceedings of the 42nd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 993–996.

Mary L. McHugh. 2012. Interrater reliability: the kappa statistic. *Biochemia medica*, 22:276–282.

Annamalai Narayanan, Mahinthan Chandramohan, Rajasekar Venkatesan, Lihui Chen, Yang Liu, and Shantanu Jaiswal. 2017. graph2vec: Learning distributed representations of graphs. *arXiv preprint arXiv:1707.05005*.

Dai Quoc Nguyen, Thanh Vu, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. 2019. A capsule network-based embedding model for knowledge graph completion and search personalization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2180–2189. Association for Computational Linguistics.

BYAMBASUREN Odmaa, Yunfei YANG, Zhifang SUI, Damai DAI, Baobao CHANG, Sujian LI, and Hongying ZAN. 2019. Preliminary study on the construction of chinese medical knowledge graph. *Journal of Chinese information proessing*, 33(10):1–7.

Liang Pang, Yanyan Lan, Jiafeng Guo, Jun Xu, Jingfang Xu, and Xueqi Cheng. 2017. Deeprank: A new deep architecture for relevance ranking in information retrieval. In *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, pages 257–266.

Chen Qu, Feng Ji, Minghui Qiu, Liu Yang, Zhiyu Min, Haiqing Chen, Jun Huang, and W Bruce Croft. 2019. Learning to selectively transfer: Reinforced transfer learning for deep text matching. In *Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*, pages 699–707.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3973–3983.

Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends in Information Retrieval*, 3(4):333–389.

Koustav Rudra and Avishek Anand. 2020. Distant supervision in bert-based adhoc document retrieval. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pages 2197–2200.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *Advances in neural information processing systems*, pages 3856–3866.

Karen Spark-Jones. 1975. Report on the need for and provision of an'ideal'information retrieval test collection. *Computer Laboratory*.

Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. Rotate: Knowledge graph embedding by relational rotation in complex space. *arXiv preprint arXiv:1902.10197*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. 2014. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, pages 1112–1119. Citeseer.

Congying Xia, Chenwei Zhang, Xiaohui Yan, Yi Chang, and Philip Yu. 2018. Zero-shot user intent detection via capsule neural networks. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3090–3099. Association for Computational Linguistics.

Liqiang Xiao, Honglun Zhang, Wenqing Chen, Yongkun Wang, and Yaohui Jin. 2018. MCapsNet: Capsule network for text with multi-task learning. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4565–4574. Association for Computational Linguistics.

Chenyan Xiong, Jamie Callan, and Tie-Yan Liu. 2017a. Word-entity duet representations for document ranking. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 763–772.

Chenyan Xiong, Zhuyun Dai, Jamie Callan, Zhiyuan Liu, and Russell Power. 2017b. End-to-end neural ad-hoc ranking with kernel pooling. In *Proceedings of the 40th International ACM SIGIR conference on research and development in information retrieval*, pages 55–64.

Zhengxin Yang, Jinchao Zhang, Fandong Meng, Shuhao Gu, Yang Feng, and Jie Zhou. 2019. Enhancing context modeling with a query-guided capsule network for document-level translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1527–1537. Association for Computational Linguistics.

Chenwei Zhang, Yaliang Li, Nan Du, Wei Fan, and Philip Yu. 2019. Joint slot filling and intent detection via capsule neural networks. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5259–5267. Association for Computational Linguistics.

Kaitao Zhang, Chenyan Xiong, Zhenghao Liu, and Zhiyuan Liu. 2020. Selective weak supervision for

neural information retrieval. In *Proceedings of The Web Conference 2020*, pages 474–485.

Wei Zhao, Haiyun Peng, Steffen Eger, Erik Cambria, and Min Yang. 2019. Towards scalable and reliable capsule networks for challenging NLP applications. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1549–1559. Association for Computational Linguistics.

Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. 2018. Investigating capsule networks with dynamic routing for text classification. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3110–3119. Association for Computational Linguistics.

Yukun Zheng, Zhen Fan, Yiqun Liu, Cheng Luo, Min Zhang, and Shaoping Ma. 2018. Sogou-qcl: A new dataset with click relevance label. In *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*, pages 1117–1120.

Yukun Zheng, Yiqun Liu, Zhen Fan, Cheng Luo, Qingyao Ai, Min Zhang, and Shaoping Ma. 2019. Investigating weak supervision in deep ranking. *Data and Information Management*, 3(3):155–164.

Andrew Zupon, Faiz Rafique, and Mihai Surdeanu. 2020. An analysis of capsule networks for part of speech tagging in high- and low-resource scenarios. In *Proceedings of the First Workshop on Insights from Negative Results in NLP*, pages 66–70. Association for Computational Linguistics.

A Appendix

A.1 The definition of four-point scale graded relevance assessment

The four-point scale graded relevance assessment (Kekäläinen, 2005) from Text REtrieval Conference (TREC) is shown in Table 5. The assessment has been adaptive for this work properly. In this work, the TREC graded relevance assessment is used as the standard to guide annotators to manually annotate the test sets and the subset from training set for Cohen's kappa coefficient computing in Section 5.1.

A.2 Constructing Training and Test sets

A.2.1 The weakly supervised training set

In this set, the relevance label between knowledge and medical literature is automatically mapped according to the *RCor* and *KImp* labels in Table 1. So each training pair is assigned a relevance label, a *RCor* label and a *KImp* label.

To calculate the *RCor* label of a pair, we first collected the *RCor* texts as described in Section 3.1, and grouped these texts based on the relationship of two entities. Since two entities will only have one relationship in CMeKG, the texts in one group are the possible candidates to describe it. Second, we replaced all the entities in the texts with a placeholder e (denoted as e-placed texts) according to CMeKG entities. Third, we used the KNN algorithm to cluster the e-placed texts, and ranked the clusters based on its text quantity. Next, we selected top 15 clusters for each group, and randomly sampled the e-placed texts in each cluster to manually check whether the texts describe the relationship. When over 70% of the sampled e-placed texts did describe, we regarded this cluster as the correct one to describe the relationship. Finally, when a new *RCor* text of one knowledge comes, the KNN cluster algorithm is used to calculate the distance with the known clusters in its group. Once the text is clustered into the correct ones, the text is regarded as describing the relationship, and the training data pair is assigned a positive *RCor* label, otherwise a negative *RCor* label.

To calculate the *KImp* label of a pair, we first used the optimized latent Dirichlet allocation (LDA) (Blei et al., 2003) model in Gensim toolkit to learn a topic model with all the medical literature. When training the LDA model, each literature is considered to discuss one independent subject. We passed the medical literature into the LDA model sequentially, and used it to construct the word frequency matrix for training. After the training, the LDA model could output the subject probability of each word in each medical literature. The word probability has been normalized, and all values add up to 1. Secondly, we used the trained LDA model to output the subject probabilities of the entities and relationship in knowledge related to one medical literature, respectively. Next, we added up the entity and relationship probabilities as the subject probability of the knowledge related to the medical literature. Finally, based on the scope of the knowledge subject probability, we set the *KImp* label at four levels, roughly corresponding to the definition in Table 1.

At last, for each knowledge and medical literature pair, we have its *RCor* label and *KImp* label, and based on the mapping definition in Table 1 we will also have an overall relevance label, which can be used to train the matching model.

A.2.2 The manually-labeled relevance prediction test set

In this set, the data pairs are randomly selected from the whole resource that excludes those in the training set. First, we recruited several professional annotators with the language skills, and they were trained in advance according to the TREC four-point scale graded relevance assessment to determine the label of a pair. These annotators knew nothing about the *RCor* and *KImp* or their definitions. Second, for each data pair, we assigned three annotators to annotate it simultaneously. Each annotator needs to assign one label from "H_r", "F_r", "M_r" and "I_r" to a data pair. Finally, we determined the label of a pair by the crowd-sourcing principle (Liu et al., 2018a). The crowd-sourcing principle is that two or more annotators give the same label, and if necessary, the third annotator will not give a label that conflicts with other annotators (Liu et al., 2018a). The label from one annotator conflicts with the other one if two labels follow one of the cases, namely ("H_r", "M_r"), ("F_r", "M_r"), and ("I_r", "H_r" or "F_r" or "M_r"). For these annotations, they will further discuss them until no conflict.

A.2.3 The manually-labeled medical literature retrieval test set

In this set, each knowledge corresponds to multiple medical literature. These medical literature is collected from the whole resource with the pooling

Table 5: The four-point scale graded relevance assessment to indicate the relevance between the medical literature and knowledge in this work.

Label	Relevance	Definition
H_r	Highly relevance	The retrieved literature discusses the themes of the knowledge exhaustively. In case of multi-faceted knowledge, all or most sub-themes or viewpoints are covered.
F_r	Fairly relevance	The retrieved literature contains more information than the knowledge description but the presentation is not exhaustive. In case of multi-faceted knowledge, only some of the sub-themes or viewpoints are covered.
M_r	Marginally relevance	The retrieved literature only points to the knowledge. It does not contain more or other information than the description.
I_r	Irrelevance	The retrieved literature does not contain any information about the knowledge.

method. The annotators need to give the relevance label between the knowledge and each medical literature. First, we randomly selected 100 knowledge from CMeKG. Second, we trained the comparison and proposed models with the training set, and then applied these trained models on knowledge to retrieval medical literature from the whole medical literature resource. Third, the popular pooling method (Spark-Jones, 1975) in IR was used, in which the top-5 literature from the retrieval results of each model was collected. All the literature (total 5,500) from these models was gathered and de-duplicated to obtain the candidate medical literature for each knowledge. Finally, the annotators manually annotated the relevance label between knowledge and its corresponding medical literature from "H_r", "F_r", "M_r" and "I_r". Therefore, in the medical literature retrieval test set, each knowledge is assigned multiple literature. The annotations and label determination process for each pair follow the same crowd-sourcing process as that in the relevance prediction test set.

The distributions of these datasets are shown in Table 6, including the numbers of each label, medical literature, knowledge, and relationship in three datasets.

A.3 Comparison Methods

***RCor&KImp* Baseline**: The baseline uses the *RCor* and *KImp* labels to automatically determine the relevance label of a pair on the relevance prediction test set or rank the candidate literature on the medical literature retrieval test set.

Unsupervised IR methods (unIR) (Robertson and Zaragoza, 2009): The unsupervised IR methods are TF · IDF and BM25 (Robertson and Zaragoza, 2009), which are popular unsupervised methods based on the term frequency (TF) and inverse document frequency(IDF) to calculate the relevance degree of knowledge and medical literature pair. On the relevance prediction test set, we set the thresholds based on the training set for different labels in both methods. On the medical literature retrieval test set, the literature is ranked based on their values.

Neural Learning-to-rank models (NeuL2R) (Xiong et al., 2017b; Dai et al., 2018; Devlin et al., 2019; Reimers and Gurevych, 2019): The general NeuL2R models include the KNRM (Xiong et al., 2017b), Conv-KNRM (Dai et al., 2018), BERT (Devlin et al., 2019), and Siamese BERT (Reimers and Gurevych, 2019), which have shown promising performance on many relevance prediction or ranking benchmarks. The KNRM and Conv-KNRM models take the medical literature and knowledge as input and output the relevance label for prediction or the relevance probability for ranking. The BERT and Siamese BERT models are pre-trained matching methods. After fine-tuned training, they can also output a relevance label for each medical literature and knowledge pair. In BERT model, the medical literature and knowledge are converted into one sequence and modeled with multi-layer Transformer architecture. The Siamese BERT model is a modification of BERT, in which the knowledge and medical literature are fed into the shared BERT encoder to learn independent representations, respectively. Two representations are fused in the last layer for the final label prediction. The source codes for KNRM, Conv-KNRM, and BERT are from the official release in GitHub. For Siamese BERT, we follow the structure described in the paper (Reimers and Gurevych, 2019) for this experiment. Besides, we have tried to implement more recent matching models (Liu et al., 2018b; Hofstätter, 2020) and BERT-based variant models (Boualili et al., 2020;

Table 6: The distributions of the medical literature (MedL), relationship (R), knowledge (K), and the pairs with different labels under the graded relevance assessment in each dataset. "*": For clarity, these data are represented with the mapped labels as shown in Table 1. There are no overlaps of medical literature or knowledge among three datasets.

Dataset	H_r	F_r	M_r	I_r	Total	MedL	R	K
Training*	14,792*	8,990*	9,249*	16,592*	49,623	36,048	97	24,160
Relevance Prediction	370	221	102	357	1,050	1,050	48	848
Ranking	1,393	857	799	420	3,469	3,469	50	100

Rudra and Anand, 2020), but we do not obtain the expected excellent results. For fair comparison, these methods are not included in this paper.

Translation based KG embedding methods (KGemb) (Bordes et al., 2013; Wang et al., 2014; Ji et al., 2015; Sun et al., 2019): The translation based knowledge graph embedding methods learn the entity or relationship representation by entity prediction. They are widely used to model the knowledge in low dimensional vector space, and also maintain the attributes of entity and relationship. First, four well-known methods are applied in this experiment, namely transE (Bordes et al., 2013), transH (Wang et al., 2014), transD (Ji et al., 2015), and rotatE (Sun et al., 2019). They are pretrained on the knowledge in CMeKG respectively, and every method could output an embedding file containing the entity embeddings and relationship embeddings. These methods are implemented from the OpenKE toolkit (Han et al., 2018). Second, in each KGemb matching model, the knowledge representation is the concatenation of the KG embedding of entities and relationship, and the medical literature presentation is from the BERT encoder. Finally, Both representations are fused in the last layer for the relevance prediction or ranking literature. Since the graph-based embedding methods, e.g. node2vec (Grover and Leskovec, 2016) and graph2vec (Narayanan et al., 2017), only focus on the node embedding in the graph and ignore the relationship, they are not included for comparison in this work.

Our implementations: In this work, first we implemented the proposed *HiCapsRKL* model according to each part description in Section 3. Second, we implemented seven additional models for the ablation study experiment. These models are CapsR$_{MedL}$, CapsR$_{RCor}$, CapsR$_{KImp}$, CapsR, RCor, KImp, and RCor&KImp. Each model is one component different from the proposed *HiCapsRKL* model.

A.4 Experimental setup:

Some experimental settings or hyper parameters in this work are listed below: The Chinese text segmentation tool for sentence processing is Jieba. During the processing, the entities in CMeKG and keywords in the literature are also added into the Jieba dictionary. The language encoder in all experiments is the BERT-base, Chinese. The max length of input pairs for modeling *RCor* and *KImp* texts is 256, and that of knowledge and medical literature pair is 512. In multi-head splitting operation, the splitting head is 12. In capsule routing method, the number of capsules is 12, and the dimension D of the input and output capsules is 64. The layer iteration K is 3. The threshold scopes of distinguishing four relevance labels in TF $\cdot$ IDF method are [0.75, 1.0), [0.6, 0.75), [0.45, 0.6), and (0.0, 0.45), respectively, and in BM25 method they are [0.6, 1.0), [0.45, 0.6), [0.35, 0.45), and (0.0, 0.35), respectively. The threshold scopes in Section A.2.1 for different *KImp* labels are [0.5, 1.0), [0.15, 0.5), [0.05, 0.15), and (0.0, 0.05), respectively.

In Section 3.1 the sentences in *KImp* text fragment are selected based on the distribution of the top-10 words in the literature. As described in Section A.2.1, we ranked and selected the top-10 words based on the subject probability of each word, and made statistics on which sentences cover the most of these top-10 words, and selected them into the KImp text fragment. Based on the coverage in the statistics, these words mostly locate in the title, keywords, first sentence, tail sentence, and other parts in descending order. The work integrates three representations in Section 3 to get the R'_{MedL} by using the "add" operation, and the cross-entropy function is the loss function in *HiCapsRKL* model training. Early stopping is used for parameter selection when training all models. All the NeuL2R, KGemb and our implemented methods are trained with the weakly supervised training data, and all comparison methods are evaluated on two test sets.