Integration of Heterogeneous Knowledge Sources for Biomedical Text Processing

Parsa Bagherzadeh and Sabine Bergler
CLaC Labs
Concordia Univesity, Montréal, Canada
{p_bagher, bergler}@cse.concordia.ca

Abstract

Recently, research into bringing outside knowledge sources into current neural NLP models has been increasing. Most approaches that leverage external knowledge sources require laborious and non-trivial designs, as well as tailoring the system through intensive ablation of different knowledge sources, an effort that discourages users to use quality ontological resources. In this paper, we show that multiple large heterogeneous KSs can be easily integrated using a decoupled approach, allowing for an automatic ablation of irrelevant KSs, while keeping the overall parameter space tractable. We experiment with BERT and pre-trained graph embeddings, and show that they interoperate well without performance degradation, even when some do not contribute to the task.

1 Introduction

Integration of external knowledge sources (KSs) is seen as a daunting task by the community. Most KSs like ontologies are large and complex. Thus, a majority of the current efforts focus on leveraging a single task relevant KS using hand-tailored architectures (Goodwin and Demner-Fushman, 2020; Peters et al., 2019; Bagherzadeh et al., 2018). During the design process, knowledge sources are often selected through an ablation study, which is laborious and makes the result task-dependent. Thus for every new task the set of relevant knowledge sources has to be identified with a similar study.

It is possible to ignore the tailoring step and use all available KSs, trusting that the training process will properly weigh the heterogeneous KSs given the internal dynamics of the model. This ideal case requires sufficient training data, but most tasks (like many biomedicial tasks) have only small or moderate-sized training data, a common problem for large, monolithic machine learning systems (Glasmachers, 2017). In those systems all KSs are always contributing their expertise, which can result in decreased rather than improved performance. We explore here a way to integrate several large, heterogeneous KSs with partly overlapping, partly divergent, and possibly even contradictory expertise in such a way that they interoperate well without adaptation, with no resulting performance decrease as well as low parameter implications. We use an integration of six KSs as our experiment system and test it over seven different shared task datasets to assess its robustness. We visualize and inspect the contribution of each KS and analyze the parameter space in detail.

The question is how to integrate multiple heterogeneous KSs so that the same system can be used for multiple, unrelated tasks without manual adaptation and without large overhead. We argue that a system with decoupled modules is suitable for this purpose. Decoupled modules can be activated conditioned on the input, allowing the system to ignore an irrelevant KS and thus preventing performance loss with fewer parameter updates at each training step (Shazeer et al., 2017). In this paradigm, instead of hand-picking KSs, an internal and automatic ablation is performed at each step for all KSs, making it easy to use the same system for different tasks, with the least detrimental effects.

Our KSs consist of the pre-trained language model BERT, as well as six structured knowledge repositories designed for human useage: WordNet, DBpedia, ConceptNet, MeSH, GO, and UMLS. For all but ConceptNet we found open source graph embeddings, and we embed ConceptNet using RDF2Vec (see Section 2).

The recently proposed multi-input RIM framework (Bagherzadeh and Bergler, 2021) comes close to our ideas and we use it here for decoupled integration of our KSs. (Bagherzadeh and Bergler, 2021) showed successful decoupled integration of

Proceedings of the 13th International Workshop on Health Text Mining and Information Analysis (LOUHI), pages 229 - 238

simple KSs like gazetteer lists that were task appropriate but did not report on experiments with large, structured KSs.

We test the same system on 7 different biomedical shared task datasets and show that our heterogeneous KSs interoperate well and achieve synergy, despite their overlap in coverage. Our results improve on two baselines contributed by the knowledge-enhanced models bioBERT (Lee et al., 2020) and KB-BERT (Hao et al., 2020). The system is competitive with state of the art systems (see Table 1).

2 Heterogeneous knowledge sources

Specialized ontological resources contain quality curated information and are often very large and complex. A graph-based knowledge representation is symbolic and discrete, making it hard to use in a machine learning framework, as most machine learning models prefer conducting computations on continuous data. The past few years have seen several techniques to embed graph structures into vector spaces. Inspired by distributional word representations (Mikolov et al., 2013), where each word is embedded in a low dimensional space, graph embedding models embed a graph into a vector space. In graph embedding models, entities (nodes) and relations (edges) are represented by vectors or matrices (Bordes et al., 2013; Ristoski and Paulheim, 2016).

Inspection of graph embedding models shows that they can capture a fair amount of ontological information. For instance (Nayyeri et al., 2021) show that related concepts are often close to each other in the vector space. We use the following ontologicial resources encoded using a pre-trained graph embedding:

WordNet is a lexical databese that defines word senses by their relations to other senses (Miller, 1995). The most important relation in WordNet is synonymy that is used to group synonymous senses into synsets.

DBpedia (Auer et al., 2007) extracts knowledge from Wikipedia info boxes, providing a large number of facts, largely focused on named entities that have Wikipedia articles.

We use the pre-trained RDF2Vec (Ristoski and Paulheim, 2016) embeddings of WordNet and DBpedia, which are available from KGvec2go website.[1]

ConceptNet is a large multi-lingual graph of general knowledge (Speer et al., 2017). ConceptNet uses closed class of 36 relations. To embed ConceptNet a set of graph embeddings is obtained in-house, using RDF2Vec.

MeSH or Medical Subject Headings is a hierarchical vocabulary, produced by the US National Library of Medicine (NLM) (Lipscomb, 2000). It is used for indexing, cataloging, and searching of biomedical and health-related information in PubMed.[2] MeSH is also embedded using a pre-trained graph embedding called MeSH2Vec (Guo et al., 2020).

GO or Gene Ontology (Ashburner et al., 2000) is a controlled vocabulary that describes gene- and protein-related terms. We use the pre-trained GO2Vec embeddings (Zhong et al., 2019) for encoding the Gene Ontology.

UMLS or Unified Medical Language System (Bodenreider, 2004) is a rich and large semantic network of biomedical vocabularies developed by NLM. UMLS comprises 127 semantic types and 54 semantic relations. Currently UMLS encompases 222 biomedical vocabularies including MeSH, GO, DrugBank, etc. For UMLS, we use the embeddings provided by (Maldonado et al., 2019).

KS	Size	Reference
WordNet	300	(Ristoski and Paulheim, 2016)
ConceptNet	200	In-House
GO	100	(Grover and Leskovec, 2016)
MeSH	64	(Guo et al., 2020)
UMLS	50	(Maldonado et al., 2019)
DBpedia	200	(Ristoski and Paulheim, 2016)

Table 1: Summary of pre-trained graph embeddings used in experiments

Table 1 provides a summary of the pre-trained graph embeddings used in the experiments. In this paper, the pre-trained graph embeddings are used off-the-shelf, without any special adjustments. We do not fine-tune the graph embeddings for three reasons. First, ontological resources represent

[1] http://kgvec2go.org
[2] https://pubmed.ncbi.nlm.nih.gov/

facts that should not be biased depending on the task. In a decoupled approach, the modules are responsible for representation learning and any task-specific adaptations are performed by the modules. Second, using graph embeddings as is enhances reproducibility of the model, as all future replications can use the same embeddings. Third, freezing the pre-trained graph embeddings significantly reduces the number of training parameters (see Section 4.5).

3 Tasks

We choose seven biomedically oriented datasets from different shared task competitions that range from simple classification tasks over multi-label classification and relation extraction to sequence labeling tasks. Comparing results for the same system on such a variety of tasks and datasets (including NER on Spanish!) allows us to be confident that the decoupled integration together with sparse activation in the miRIM architecture successfully avoids interference of the KSs and performance degradation.

BB-Rel or Bacteria Biotope which is part of the BioNLP 2019 challenge focuses on the extraction of two types of relations namely `Lives_In` and `Exhibits` (Bossy et al., 2019). `Lives_In` relations link a microorganism entity to its location. `Exhibits` relations on the other hand link a microorganism entity to a phenotype entity. To evaluate the test predictions we use the online tool provided by the organizers.[3]

ChemProt or BioCreative VI track 5 involves detection of relations between mentions of chemicals and genes/proteins in medical journals (Krallinger et al., 2017). The ChemProt task provides a manually annotated corpus, where domain experts have exhaustively labeled all chemical and gene mentions, and all binary interactions between them corresponding to a specific set of biologically relevant relation types, called ChemProt relation classes (CPRs).

DDI or SemEval 2013 task 9.b (Segura-Bedmar et al., 2013) is a relation extraction task for drug-drug interaction mentions in DrugBank (Wishart et al., 2018) and MedLine abstracts.

[3] `http://bibliome.jouy.inra.fr/demo/BioNLP-OST-2019-Evaluation/index.html`

HoC or Hallmarks of Cancer (Baker et al., 2015) is a multi-label classification task where zero or more labels are assigned to sentences from PubMed abstracts describing cancer hallmarks. Note that the HoC data set is not pre-spitted into train, development, and test sets. We therefore randomly split the data with 60%, 20%, and 20% ratios for train, development, and test respectively.

LitCov or BioCreative VII track 5 (Chen et al., 2021) concerns multi-label classification of abstracts from Covid-related articles into 7 classes, namely: *Treatment*, *Mechanism*, *Prevention*, *Case Report*, *Diagnosis*, *Transmission*, and *Epidemic Forecasting*.

LivNER is a sequence labeling task that requires recognition and classification living things into the two categories HUMAN and SPECIES in Spanish clinical reports. Note that since LivNER is a recent challenge, the gold standard labels for the official test set is not disclosed, thus, we used a hold out test set from the training data.

PPI or BioCreative III Article Classification Task (ACT) is a binary task in which biomedical articles describing protein-protein interactions (PPI) must be identified (Krallinger et al., 2011).

Task	Metric	Train/Dev/Test split
BB-Rel	F1	1000/64/500
ChemProt	F1	1682/612/800
DDI	F1	500/214/191
HoC	F1	10.4k/3.5k/3.5k
LitCov	mac-F1	24.9k/6.2k/2.5k
LivingNER	μF1	500/250/250
PPI	Acc	6280/6000

mac-F1: macro-F1, **μF1**: micro-F1, **Acc**: Accuracy

Table 2: Size and evaluation metric for datasets

Table 2 provides a summary of the biomedical tasks. The tasks differ in their complexity, number of training samples, as well as the type of knowledge they require. The diversity of the biomedical tasks allows to evaluate the efficacy of the decoupled integration of heterogeneous KSs.

4 Experiments

4.1 Decoupled framework

As described in (Bagherzadeh and Bergler, 2021), mi-RIM is an architecture of M decoupled recurrent modules $f_1, \ldots, f_M$, where each module f_m operates on a different input, making it possible to integrate different KSs.

In mi-RIM, each KS_m (for instance a pretrained model) provides its representation x_t^m for a token at position t to the module f_m. Module f_m selects its input using an attention mechanism:

$$\tilde{x}_t^m = Attention(h_{t-1}^m, X_t^m, X_t^m) \quad (1)$$

where $Attention(h_{t-1}^m, X_t^m, X_t^m)$ is the dot-product attention (Vaswani et al., 2017) with h_{t-1}^m as query and X_t^m as both key and value, and $X_t^m = [\mathbf{0}\,; x_t^m]$, where $\mathbf{0}$ is an all-zero vector and ; denotes row-level concatenation. This attention mechanism allows a module to ignore the input from a KS by attending more to the null input (the all-zero vector).

Once all modules have selected their input, M sets of attention scores are available. Among the modules, a set of top-k modules with the least attention to the null input are selected as active modules, denoted by $\mathcal{F}_t$. As argued by (Goyal et al., 2019), sparse activity leads to competition among modules which leads to developing more specialized expertise for them. We show that this input selection mechanism allows for an automatic ablation of KSs, identifying and blocking irrelevant ones and thus preventing a module to be updated by its corresponding KS.

The active modules are updated using their selected input to obtain temporary hidden representations $\tilde{h}_t^m$ ($m \in F_t$):

$$\tilde{h}_t^m = f_m(\tilde{x}_t^m, h_{t-1}^m) \quad m \in \mathcal{F}_t \quad (2)$$

where $f_m(\tilde{x}_t^m, h_{t-1}^m)$ denotes a single recurrence of f_m with $\tilde{x}_t^m$ as input and h_{t-1}^m as previous hidden state. For the inactive modules, the temporary hidden representation is copied from the previous position, in other words, $\tilde{h}_t^m = h_{t-1}^m$.

The active modules then interact with each other via another attention mechanism to obtain their actual hidden representations:

$$h_t^m = Attention(\tilde{h}_t^m, \tilde{H}_t, \tilde{H}_t) \quad m \in \mathcal{F}_t \quad (3)$$

where $\tilde{H}_t = [\tilde{h}_t^1; \ldots; \tilde{h}_t^M]$. The actual hidden state for inactive modules is the same as their temporary hidden state ($h_t^m = \tilde{h}_t^m \; m \notin F_t$).

Because the input selection and interaction mechanisms are attention based and attention can take a variable number of argument representation, new KS modules can be added to an existing model without major changes.

(Bagherzadeh and Bergler, 2021) provided a proof of concept for integration of language models and a few gazetteer lists on simple tweet-related biomedical tasks. Here, instead, we test the decoupled mi-RIM framework on complex tasks and on a more diverse set of KSs.

4.2 Preprocessing and implementation details

We use a GATE pipeline (Cunningham et al., 2002) for preprocessing with CoreNLP (Manning et al., 2014) plugin for tokenization and sentence splittting. For LivNER we use the Spanish version of CoreNLP for preprocessing[4]. The integration of KSs requires minimal prepossessing. Tokens are matched against each ontology using a simple case-insensitive exact match approach, by matching for the longest possible span. The exact matching approach is widely used for incorporating external KSs. For instance (Goodwin and Demner-Fushman, 2020) successfully use exact matching to incorporate information from ConceptNet.

We use the PyTorch library (Paszke et al., 2017) for mi-RIM implementation. We use 7 LSTM modules[5] to accommodate the BERT and the graph embeddings. The hidden size of all modules is set to $d_h = 128$. All models are trained using the Adam optimizer (Kingma and Ba, 2015) with a learning rate of $lr = 5 \times 10^{-6}$.

4.3 Numerical results

Table 3 reports the results. The first 2 rows of the table report the performance of BERT and BERT (frozen) as the sole KSs, forming baselines for the experiments. We use the same system with all knowledge sources for all tasks to observe how the system behaves for widely different tasks with different knowledge requirements. For LivNER,

[4] see https://stanfordnlp.github.io/CoreNLP/human-languages.html

[5] see https://pytorch.org/docs/stable/generated/torch.nn.LSTMCell.html

KSs	M	k	mic-F1 BB-Rel	mic-F1 ChemProt	F1 DDI	mac-F1 LitCov	F1 PPI	F1 HoC	mic-F1 LivNER
BERT (frozen)	1	1	58.3	68.3	85.7	75.5	68.3	79.1	85.3
BERT	1	1	62.9	74.2	87.3	79.2	70.2	83.1	87.9
BERT (frozen), All Graph Emb.	7	7	64.1	70.9	87.2	79.2	72.6	82.4	88.9
		6	64.7	71.4	87.8	79.7	73.1	82.7	89.3
		5	64.9	72.2	88.3	80.6	73.8	83.4	89.5
		4	66.0	73.4	88.6	81.1	**74.2**	83.8	90.4
		3	**66.1**	**74.1**	**88.9**	**81.3**	73.3	**84.3**	**90.6**
		2	64.7	72.6	87.6	79.1	72.6	82.2	89.4
		1	62.9	70.1	86.3	76.7	70.9	81.6	88.8
BERT, All Graph Emb.	7	7	66.3	76.2	89.8	83.2	73.8	85.6	91.2
		6	66.9	76.8	90.1	84.2	74.1	85.9	91.5
		5	67.4	77.0	90.7	84.7	74.6	86.3	91.8
		4	**67.6**	77.4	91.3	85.1	**75.7**	86.5	92.3
		3	66.8	**78.8**	**91.8**	**85.6**	74.9	**86.9**	**92.8**
		2	66.2	77.3	89.0	83.0	73.1	85.2	91.6
		1	64.5	75.5	88.2	81.9	72.2	84.4	89.7
BioBERT			65.3	75.2	89.9	81.7	73.8	84.6	NA
KB-BERT			65.8	76.1	90.3	81.5	72.7	85.1	NA
SOTA			64.8[1]	77.2[2]	92.2[3]	88.7[4]	NA	NA	NA

1. (Zhang et al., 2019) 2. (Gu et al., 2021) 3. (Luo et al., 2020) 4. (Fang and Wang, 2021)

Table 3: Decoupled Integration of KSs using a mi-RIM. The same system is used for all tasks

which is a Spanish task, we use the Spanish version of BERT (Cañete et al., 2020).

The table indicates the number of modules M: for the simple BERT baseline, there is only one module (namely, BERT). The experimental system has 7 modules, 6 graph embeddings and BERT's language model.

Column 3 indicates the degree of enforced sparsity k. When all modules are active ($k = 7$), all KSs contribute their information and update their corresponding modules. In this case, the system corresponds to a monolithic model and shows small improvements (of 2-5%). This shows that to some extent, the monolithic model can ignore irrelevant KSs using its inner dynamics.

Integration of BERT with the extant graph embeddings never shows loss of performance compared to the respective baseline. This is a strong result, considering how heterogeneous the KSs are and how varied and small the datasets and tasks. The strongest results of the decoupled design with sparse activation are for $k = 3$ or $k = 4$. This can be attributed to the competition among KSs, allowing for their contribution only if they are relevant to the task using input selection. This mitigates the inclusion of irrelevant KSs. For instance, LitCov and DDI tasks do not require knowledge on genes, thus GO is an irrelevant KS. Nevertheless, its inclusion does not lead to a performance decrease for the two tasks compared to the BERT and BERT (frozen) baselines. They are, however, the only two tasks for which SOTA outperforms our experimental system for $k = 3$.

Extreme sparsity ($k = 1$ or $k = 2$) shows lower performance than $k = 3$ but never below the BERT baselines. $k = 1$ is generally lower than $k = 7$ but still close to SOTA performances. This shows that although the system is forced to ablate most of the KSs, it can still find a combination that improves overall performance. Note that $k = 1$ is not equivalent to injecting only a single KS into the system since the miRIM architecture makes decisions at the token level and in certain cases the computation graph for $k = 1$ may include all 7 KSs.

As discussed in the next section, sparsity generally leads to a significant reduction in the number of parameters.

Although LitCov (which has the largest training set) benefits the most from the integration of

KSs compared to its BERT baseline, other tasks with smaller sized training data also show sizable improvements, which are more pronounced with sparse activity of the modules. This demonstrates the benefits of an automatic internal ablation mechanism for integration of large heterogeneous KSs.

In general, a decoupled approach also allows to reuse embeddings of KSs. Consider LivNER, which is a Spanish task. We use the same system as for the English tasks and only replace BERT with its Spanish version. Note that a language model trained on Spanish text has significantly different representations compared to its English version, however, as the results suggest, it interoperates well with the other (English) KSs. This recommends the approach also for underresourced languages.

The pre-trained graph embeddings also interoperate well with frozen BERT. The results show that once integrated with frozen BERT (which has no fine tuning on the target task datasets), the lexical information in the knowledge sources effectively compensates for the loss. In most cases, integration of the off-the-shelf KSs with frozen BERT outperforms fine-tuned BERT significantly with almost 100M less parameters. This is very attractive for training on small or moderated-sized data, with less potential for overfitting (Li et al., 2021) or in resource limited situations.

Table 3 also reports on other knowledge enhanced models such as BioBERT (Lee et al., 2020) and KB-BERT (Hao et al., 2020), as well as the state-of-the-art (SOTA). With sparse activity ($k = 4$ or $k = 3$), integration of lexical KSs with BERT always outperforms both BioBERT and KB-BERT, showing that the automatic ablation of discrete KSs is competitive with domain specific pre-training.

Note that the k values for best-preforming settings fall within an arrow interval ($k = 3$ or $k = 4$), suggesting that automatic mechanisms can be used to determine k during training.

4.4 Analysis of results

In precision-oriented applications such as biomedical tasks, users require to understand why and how a prediction is made (Amini and Kosseim, 2019). In a decoupled approach, the activity of each module is often transparent for inspection. Likewise, in mi-RIM, contributions of KS are transparent. Each module selects its input from its corresponding KS using an attention mechanism and if the input is deemed relevant, the module has a high chance of activation. The activation patterns can be traced, providing insight into the functionality of the system. Consider Example 1 (from HoC task):

(1) *Unlike insulin, ghrelin inhibited Akt kinase activity as well as up-regulated gluconeogenesis*

In this example, the term *gluconeogenesis* is matched with UMLS, MeSH, GO, ConceptNet, and DBpedia. Note that BERT also provides a representation for the term. Figure 1 shows the activation patterns of mi-RIM for Example 1. The gray regions indicate activity for a module.

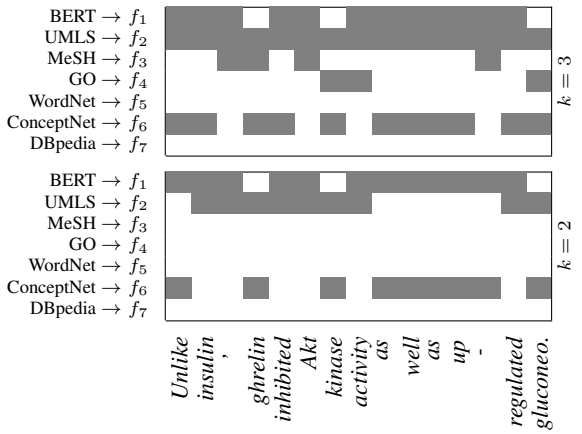

Figure 1: Activation patterns of mi-RIM for Example 1

For the term *gluconeogenesis*, when $k = 3$, modules f_2, f_4, and f_6 (corresponding to UMLS, GO, and ConceptNet respectively) win the competition and are active. Note that the model has selected a very specialized KS for genes (GO), a more comprehensive KS (UMLS), and a general KS (ConceptNet). This suggests that the model is trying to balance the expertise of active KSs. In this light, the activity of ConceptNet versus the inactivity of MeSH is interesting where the general resource ConceptNet is selected over the more specialized MeSH. A similar pattern is also observed when $k = 2$, where ConceptNet is selected over GO, suggesting that it is a more robust resource.

The activation patterns suggest that an automatic and internal ablation is performed by the decoupled model. This suggests that an established system of M KSs can be used for different tasks

without pre-ablating relevant KSs because contributions of irrelevant KSs are mitigated by input selection.

4.5 Parameter space and inference time

Let Θ_{mod} denote the set of training parameters implicated by all modules and $|\Theta_{mod}|$ denotes the overall number of parameters. Due to conditional computation in mi-RIM, the number of trained parameters $|\Theta_{mod}|'$ (sample-wise) is linked to the value of k. If $k = M$, all modules are part of the computation graph, i.e. all parameters are trained: $|\Theta_{mod}|' = |\Theta_{mod}|$.

However, when $k < M$ (sparse activity) $\frac{k}{M}|\Theta_{mod}| \leq |\Theta_{mod}|' \leq |\Theta_{mod}|$. The best case ($\frac{k}{M}|\Theta_{mod}|$) occurs when $M-k$ modules are never active and thus not included in the computation graph. The worst case ($|\Theta_{mod}|$) on the other hand occurs, when all modules are active at least for a single position t, forcing all to be included in the computation graph.

Consider the activation patterns of Figure 1 when $k = 3$. Module f_4 (corresponding to GO) is active only at three positions, leading to the inclusion of the module in the computation graph. Moreover, module f_3 (corresponding to MeSH) shows activity for four positions. Although the top-k activity is set to 3, overall, 5 modules demonstrate activity for at least one position. In this case, $|\Theta_{mod}|' = \frac{5}{7}|\Theta_{mod}|$. Note that the best case when $k = 3$, is $|\Theta_{mod}|' = \frac{3}{7}|\Theta_{mod}|$. Although more reduction is expected with smaller values of k, it is possible that all modules demonstrate activity at least for one position even if $k = 1$.

Figure 2 shows a comparison of the fraction of trained parameters $\frac{|\Theta_{mod}|'}{|\Theta_{mod}|}$ for two different tasks. Sparse activity consistently reduces the number of trained parameters. Note that on average, the fraction of trained parameters never approaches its best case ($\frac{k}{M}$). For instance, when $k = 1$, for HoC, $\frac{|\Theta_{mod}|'}{|\Theta_{mod}|} = 0.46$ while the best case is about 0.14. This shows that on average 3.2 modules show activity at least for one position even tough $k = 1$.

The reported experiments showed that most runs demonstrate their best performance when $k = 4$ or 3. As Figure 2 shows, on average, when $k = 4$ and $k = 3$, 67% and 52% of parameters are trained respectively. This shows that while improving performance, sparse activity can significantly reduce the number of trained parameters.

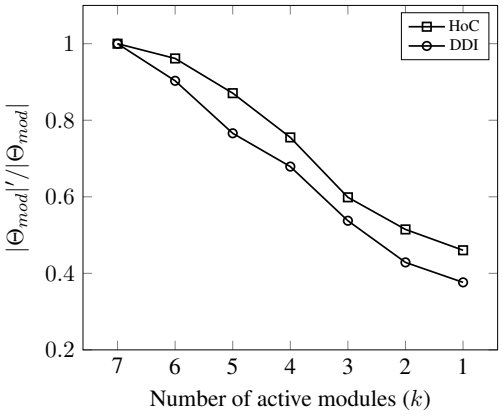

Figure 2: Fraction of trained parameters vs number of active modules

The reduced parameter space allows for training on small or moderate-sized data sets with less potentials for over-fitting (Li et al., 2021).

A brief analysis of the inference time is also provided in Figure 3. We measure the inference time for different values of top-k activity. Note that the reported inference time is the average timing on all tasks, timed on an Intel Corei7 CPU.

As Figure 3 shows, sparse activity significantly reduces the inference time. This is expected since once a KS is not selected, there is no need to update its corresponding module, leading to speed-up in the inference time.

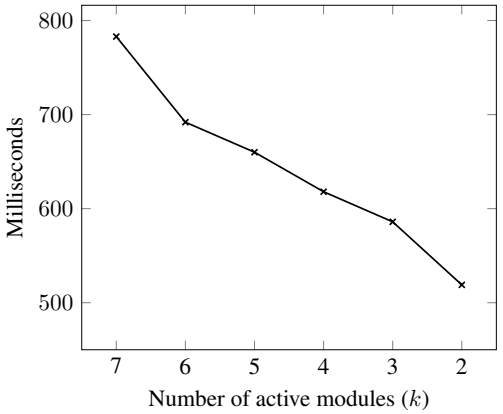

Figure 3: Inference time of mi-RIM with 7 KSs/modules for different k values

5 Conclusion

This paper presents extensive experiments on decoupled integration of heterogeneous KSs such as language models and pre-trained graph embeddings. The same system with all KSs was used for all tasks, without special calibrations, demon-

strating reusability of extant knowledge sources.

The tasks differed in terms of complexity as well as their knowledge requirements (specialized or general knowledge). The results show that for the tasks considered here, the KSs interoperate well and they do not confound each other's performances. Moreover, we showed that a system that leverages multiple KS does not necessarily show significant improvement, rather the sparse activity of modules is required to effectively improve performance.

Inspection of activation patterns shows that a decoupled system can ignore irrelevant/redundant KSs, showing an automatic ablation behavior.

We show that in terms of the number of trained parameters, a decoupled approach is efficient. The sparse activity significantly reduces the number of trained parameters. Moreover, since the pre-trained graph embeddings are not fine-tuned, the overall model does not have large parameter implications.

We also stress the ease of reusing and replicating such a decoupled system, since the same pre-trained embeddings will be used by different users. Moreover, the pre-trained embeddings do not have to be stored on the same machine that the model is trained on. KGvec2go[6], for instance, provides an API through which pre-trained embeddings are accessible. This ultimately results in lightweight models.

In conclusion, a decoupled approach allows for robust and efficient integration of heterogeneous KSs, allowing the user to leverage multiple knowledge sources, without any need for special calibration or tailoring.

References

Hessam Amini and Leila Kosseim. 2019. Towards explainability in using deep learning for the detection of anorexia in social media. In *Natural Language Processing and Information Systems*, pages 225–235. Springer International Publishing.

Michael Ashburner, Catherine A Ball, Judith A Blake, David Botstein, Heather Butler, J Michael Cherry, Allan P Davis, Kara Dolinski, Selina S Dwight, Janan T Eppig, et al. 2000. Gene ontology: tool for the unification of biology. *Nature genetics*, 25(1):25–29.

Sören Auer, Christian Bizer, Georgi Kobilarov, Jens Lehmann, Richard Cyganiak, and Zachary Ives. 2007. Dbpedia: A nucleus for a web of open data. In *The semantic web*, pages 722–735. Springer.

Parsa Bagherzadeh and Sabine Bergler. 2021. Multi-input Recurrent Independent Mechanisms for leveraging knowledge sources: Case studies on sentiment analysis and health text mining. In *Proceedings of Deep Learning Inside Out (DeeLIO): The 2nd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 108–118.

Parsa Bagherzadeh, Nadia Sheikh, and Sabine Bergler. 2018. CLaC at SMM4H task 1, 2, and 4. In *Proceedings of the 2018 EMNLP Workshop SMM4H: The 3rd Social Media Mining for Health Applications Workshop & Shared Task*.

Simon Baker, Ilona Silins, Yufan Guo, Imran Ali, Johan Högberg, Ulla Stenius, and Anna Korhonen. 2015. Automatic semantic classification of scientific literature according to the hallmarks of cancer. *Bioinformatics*, 32(3):432–440.

Olivier Bodenreider. 2004. The Unified Medical Language System (UMLS): Integrating Biomedical Terminology. *Nucleic acids research*, 32(suppl_1):D267–D270.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating Embeddings for Modeling Multi-Relational Data. In *Neural Information Processing Systems (NIPS)*, pages 1–9.

Robert Bossy, Louise Deléger, Estelle Chaix, Mouhamadou Ba, and Claire Nédellec. 2019. Bacteria biotope at BioNLP open shared tasks 2019. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*, pages 121–131.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish Pre-Trained BERT Model and Evaluation Data. In *PML4DC at ICLR 2020*.

Qingyu Chen, Alexis Allot, and Zhiyong Lu. 2021. Litcovid: an open database of covid-19 literature. *Nucleic acids research*, 49(D1):D1534–D1540.

Hamish Cunningham, Diana Maynard, Kalina Bontcheva, and Valentin Tablan. 2002. GATE: A Framework and Graphical Development Environment for Robust NLP tools and applications. In *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*.

Li Fang and Kai Wang. 2021. Team bioformer at biocreative vii litcovid track: Multic-label topic classification for covid-19 literature with a compact bert model. In *Proceedings of the seventh BioCreative challenge evaluation workshop*.

Tobias Glasmachers. 2017. Limits of end-to-end learning. In *Asian Conference on Machine Learning*, pages 17–32.

[6] http://kgvec2go.org/

Travis Goodwin and Dina Demner-Fushman. 2020. Enhancing question answering by injecting ontological knowledge through regularization. In *Proceedings of Deep Learning Inside Out (DeeLIO): The First Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 56–63.

Anirudh Goyal, Alex Lamb, Jordan Hoffmann, Shagun Sodhani, Sergey Levine, Yoshua Bengio, and Bernhard Schölkopf. 2019. Recurrent independent mechanisms. *arXiv preprint arXiv:1909.10893*.

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.

Yu Gu, Robert Tinn, Hao Cheng, Michael Lucas, Naoto Usuyama, Xiaodong Liu, Tristan Naumann, Jianfeng Gao, and Hoifung Poon. 2021. Domain-specific language model pretraining for biomedical natural language processing. *ACM Transactions on Computing for Healthcare (HEALTH)*, 3(1):1–23.

Zhen-Hao Guo, Zhu-Hong You, De-Shuang Huang, Hai-Cheng Yi, Kai Zheng, Zhan-Heng Chen, and Yan-Bin Wang. 2020. MeSHHeading2vec: a new method for representing MeSH headings as vectors based on graph embedding algorithm. *Briefings in Bioinformatics*, 22(2):2085–2095.

Boran Hao, Henghui Zhu, and Ioannis Paschalidis. 2020. Enhancing clinical BERT embedding using a biomedical knowledge base. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 657–661.

Diederik P Kingma and Jimmy Lei Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of the 3rd International Conference on Learning Representations, ICLR'15*.

Martin Krallinger, Obdulia Rabal, Saber A Akhondi, Martın Pérez Pérez, Jesús Santamaría, GP Rodríguez, et al. 2017. Overview of the biocreative vi chemical-protein interaction track. In *Proceedings of the sixth BioCreative challenge evaluation workshop*, volume 1, pages 141–146.

Martin Krallinger, Miguel Vazquez, Florian Leitner, David Salgado, Andrew Chatr-Aryamontri, Andrew Winter, Livia Perfetto, Leonardo Briganti, Luana Licata, Marta Iannuccelli, et al. 2011. The protein-protein interaction tasks of biocreative iii: classification/ranking of articles and linking bio-ontology concepts to full text. *BMC bioinformatics*, 12(8):1–31.

Jinhyuk Lee, Wonjin Yoon, Sungdong Kim, Donghyeon Kim, Sunkyu Kim, Chan Ho So, and Jaewoo Kang. 2020. Biobert: a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4):1234–1240.

Zhu Li, Zhi-Hua Zhou, and Arthur Gretton. 2021. Towards an understanding of benign overfitting in neural networks. *arXiv preprint arXiv:2106.03212*.

Carolyn E Lipscomb. 2000. Medical subject headings (MeSH). *Bulletin of the Medical Library Association*, 88(3).

Ling Luo, Zhihao Yang, Mingyu Cao, Lei Wang, Yin Zhang, and Hongfei Lin. 2020. A neural network-based joint learning approach for biomedical entity and relation extraction from biomedical literature. *Journal of biomedical informatics*, 103:103384.

Ramon Maldonado, Meliha Yetisgen, and Sanda M Harabagiu. 2019. Adversarial learning of knowledge embeddings for the unified medical language system. *AMIA Summits on Translational Science Proceedings*, 2019:543.

Christopher D. Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven J. Bethard, and David McClosky. 2014. The Stanford CoreNLP natural language processing toolkit. In *Association for Computational Linguistics (ACL) System Demonstrations*, pages 55–60.

Tomas Mikolov, Wen-tau Yih, and Geoffrey Zweig. 2013. Linguistic regularities in continuous space word representations. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 746–751.

George A Miller. 1995. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41.

Mojtaba Nayyeri, Sahar Vahdati, Can Aykul, and Jens Lehmann. 2021. 5* knowledge graph embeddings with projective transformations. In *AAAI 2021*. AAAI Press.

Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. 2017. Automatic differentiation in PyTorch. In *Neural Information Processing Systems (NIPS)*.

Matthew E. Peters, Mark Neumann, Robert Logan, Roy Schwartz, Vidur Joshi, Sameer Singh, and Noah A. Smith. 2019. Knowledge enhanced contextual word representations. 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 43–54.

Petar Ristoski and Heiko Paulheim. 2016. RDF2Vec: RDF graph embeddings for data mining. In *International Semantic Web Conference*, pages 498–514. Springer.

Isabel Segura-Bedmar, Paloma Martínez, and María Herrero-Zazo. 2013. SemEval-2013 task 9 : Extraction of drug-drug interactions from biomedical texts

(DDIExtraction 2013). In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, pages 341–350.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer. *arXiv preprint arXiv:1701.06538*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *"Proceedings of AAAI 2017 Conference on Artificial Intelligence*, pages 4444–4451.

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.

David S Wishart, Yannick D Feunang, An C Guo, Elvis J Lo, Ana Marcu, Jason R Grant, Tanvir Sajed, Daniel Johnson, Carin Li, Zinat Sayeeda, et al. 2018. Drugbank 5.0: a major update to the drugbank database for 2018. *Nucleic acids research*, 46(D1):D1074–D1082.

Qi Zhang, Chao Liu, Ying Chi, Xuansong Xie, and Xiansheng Hua. 2019. A multi-task learning framework for extracting bacteria biotope information. In *Proceedings of The 5th Workshop on BioNLP Open Shared Tasks*.

Xiaoshi Zhong, Rama Kaalia, and Jagath C Rajapakse. 2019. Go2vec: transforming go terms and proteins to vector representations via graph embeddings. *BMC genomics*, 20(9):1–10.