Re-Representation in Sentential Relation Extraction with Sequence Routing Algorithm

Ramazan Ali Bahrami and **Ramin Yahyapour**

Georg-August-Universität Göttingen and GWDG , Göttingen, Germany

`{ramazan.bahrami, ramin.yahyapour}@gwdg.de`

Abstract

Sentential relation extraction (RE) is an important task in natural language processing (NLP). In this paper we propose to do sentential RE with dynamic routing in capsules. We first show that the proposed approach outperform state of the art on common sentential relation extraction datasets Tacred, Tacredrev, Retacred, and Conll04. We then investigate potential reasons for its good performance on the mentioned datasets, and yet low performance on another similar, yet larger sentential RE dataset, Wikidata. As such, we identify noise in Wikidata labels as one of the reasons that can hinder performance. Additionally, we show associativity of better performance with better re-representation, a term from neuroscience referred to change of representation in human brain to improve the match at comparison time. As example, in the given analogous terms King:Queen::Man:Woman, at comparison time, and as a result of re-representation, the similarity between related head terms (King,Man), and tail terms (Queen,Woman) increases. As such, our observation show that our proposed model can do re-representation better than the vanilla model compared with. To that end, beside noise in the labels of the distantly supervised RE datasets, we propose re-representation as a challenge in sentential RE [1].

1 Introduction

Sentential relation extraction is about inferring the relation between two entities in a given sentence. Various sentential relation extraction datasets are constructed with distant supervision. Accordingly, it is assumed that if two entities are related in a knowledge base such as Wikidata, they are also related according to the sentence that contain them. As such, datasets often not only provide sentences with corresponding entities, but also additional details such as description, aliases, and types of entities. Accordingly, to improve performance, various
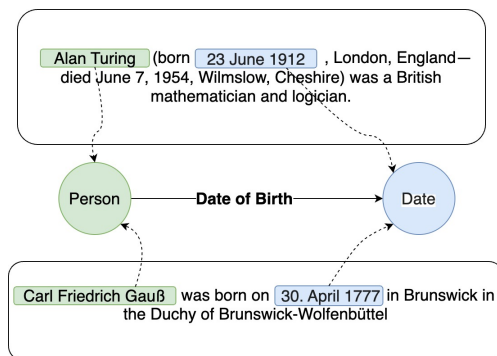

Figure 1: The need for re-representation in sentential relation extraction (Text from Wikipedia). At comparison time, the similarity between related terms increases. Note the entity types as manual label for re-representation. Note also the proportional word analogy constructed from the given example here, "Alan Turing": 1912 :: "C.F.GauSS":1777

works introduce complicated models that account for extra additional details. To that end, the results of incorporating additional details are not coherent. As example, while in some studies entity type has shown to improve performance (Bastos et al., 2021), in some others however, it has shown to degrade performance(Vashishth et al., 2018). As such, weather there are still room for improvement or if low performance are due to noise and error in the labels, is an open research question. In this paper, we aim to offer a deeper understanding of the task that can help in defining a better goal and objective for methods that incorporate the additional details into the sentential context using complex models. To do so, we propose an intuitive model that out perform state of the art on most dataset, and then investigate the reason for its better performance. To propose our approach, we build on works from neuroscience. To that end, inference such as sentential relation extraction in which a relation from one context, is mapped to a relation in another context is referred to as analogical reasoning (Gentner, 1983). In analogies, such as proportional word

---

analogies of the form King:Queen::Man:Woman, 'King' is related to 'Queen' as 'Man' is related to 'Woman', even though pairs (King,Man) or heads and (Queen,Woman) or tails are different. Similarly in sentential relation extraction, given that according to sentences S_i, for $i \in \{1, 2, ..., N\}$; the head entity, e_i^h and tail entity, e_i^t are similarly related, we have analogous terms of the form $e_i^h : e_i^t :: e_j^h : e_j^t$. To that end, studies in neuroscience suggest that, comparison as the foundation of any analogical reasoning, changes the representation of objects being compared (Gentner and Kurtz, 2006; Lorenza Saitta, 2013). As example in (Silliman and Kurtz, 2019) empirical evidence of re-representation, according to which people changes the representation of entities in order to improve the match at comparison time is documented. In other words, given that analogies are about partial similarity in different contexts (Hummel and Doumas, 2023), to map a relation from one context to a relation in another context, it is therefore needed to discard some information in both contexts (Lorenza Saitta, 2013) Figure 1. Accordingly, we found dynamic routing in Capsules network suitable for the task. Capsules were introduced first by (Hinton et al., 2011), and dynamic routing in capsules by (Sabour et al., 2017). They can be thought of neurons that output different features of processed entity. We test the proposed algorithm on relation extraction datasets Wikidata (Sorokin and Gurevych, 2017), Tacred (Zhang et al., 2017), Tacredrev (Alt et al., 2020), Retacred (Stoica et al., 2021), and Conll04 (Roth and Yih, 2004; Markus and Adrian, 2020). Our observations are summarized as follows:

- Our proposed approach improve state of the art scores on sentential relations extraction datasets Tacred, Tacredrev, Retacred, and Conll04.

- We estimate a significant error rate in labels of Wikidata, the dataset on which various studies try to improve model performance by incorporating the additional details through complex models.

- We show empirical evidence of re-representation and its associativity with better sentential RE performance in neural network.

2 Related Works

The use of extra additional details about entities such as entity type, aliases and description, and the way they are incorporated into the sentential context is one of the main theme of related works. As such, beside studies that addresses noise in the RE datasets, the other works deal with extra additional details and how to best incorporate them in the context. To that end, (Riedel et al., 2010) show that the vanilla distant supervised method used for generating sentential RE datasets, result in noisy labels, and proposes an improved version of the vanilla method, reducing error rate by 30%. Additionally, in studies related to variants of the common sentential RE dataset Tacred (Zhang et al., 2017), Tacredrev (Alt et al., 2020), and Retacred (Stoica et al., 2021); it is shown that after relabeling the noisy examples in Tacred, models improve performance by 8.0% (Tacredrev) and 14.3% (Retacred) of F1 score. Moreover, state of art performance for Tacred, and its variants, is proposed by (Zhou and Chen, 2022; Park and Kim, 2021). They show that incorporating abstract label of entities(entity types) improve model performance. Furthermore, (Sorokin and Gurevych, 2017) introduces Wikidata, a much larger dataset for sentential relation extraction based on the knowledge base wikidata (Table 1). To that end, to improve performance by enriching sentential context, in addition to entity type, (Nadgeri et al., 2021; Bastos et al., 2021) consider integrating other side information such as entity description, and aliases, through complex models such as graph neural network. Moreover, in (Vashishth et al., 2018), the use of entity types and relation alias information for improving performance is discussed.

3 Problem Formulation

We formulate the sentential RE based on the fundamental assumption that it is a type of analogical reasoning. To the end, in relation extraction datasets, according to sentences S_i, and S_j we can construct proportional word analogies of the form $e_i^h : e_i^t :: e_j^h : e_j^t$, as we have King:Queen::Man:Woman. As such, a claim based on the studies from neuroscience is that our ideal proposed model shall do re-representation(Silliman and Kurtz, 2019). More commonly, given $X_i^h : X_i^t :: Y_j^h : Y_j^t$, re-representation can be viewed equivalent to a transformation F such that according to some similarity measures ψ, when X_i^h is related to X_i^t, as Y_j^h is
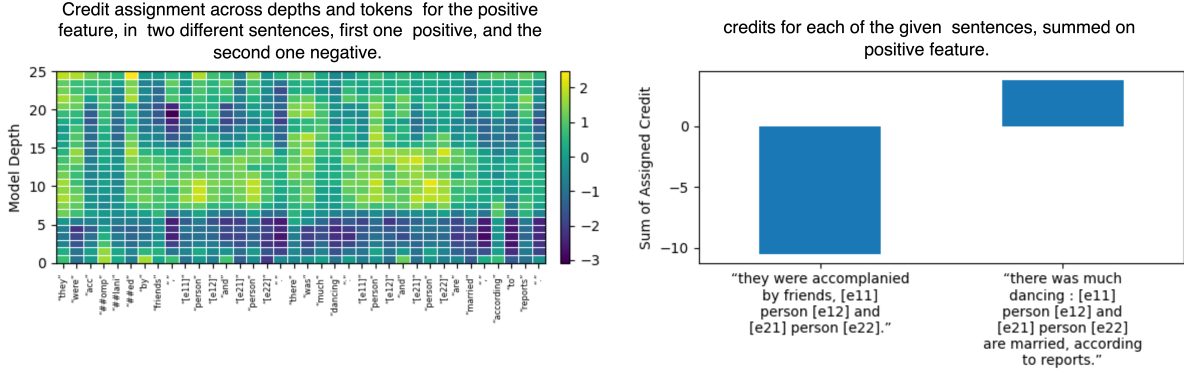
Figure 2: Credit assignment in dynamic routing (Heinsen, 2022). The output here has 2 dimensions only, one for positivity, and the other for negativity. As example, given some sequence of vectors of depth h (25 here), sequence number n (number of tokens in the respective sentence), and dimension d (1024 here), and some configuration for the expected output (here depth=1, d=2, n=1), the dynamic routing algorithm works as credit assignment system. As such, projections of every feature in the input has limited credits at their disposal, and assigns it to the features in the output. Summing credits over all hidden states for positive feature, result in a value that is greater when the example is positive, and smaller otherwise(Example sentences are taken from Retacred).

related to Y_j^t (positive examples) , we have :

$$\psi(F(X_i^h), F(Y_j^h)) \simeq 1$$
$$\psi(F(X_i^t), F(Y_j^t)) \simeq 1$$

and when X_i^h is not related to X_i^t, as Y_j^h is related to Y_j^t (negative examples), we have:

$$\psi(F(X_i^h), F(Y_j^h)) \simeq -1$$
$$\psi(F(X_i^t), F(Y_j^t)) \simeq -1$$

As such, the similarity function ψ returns 1 when terms come from positive examples and -1 otherwise. One common example for ψ is cosine similarity between two given vectors. It is to note that, as in sentential RE, the sentences containing entities express the relation between entities, it is therefore needed that any change of representation be conditioned on the contextual sentence. As such, formally the task can be presented as the minimization problem below:

$$\min_{t \in [s,o]} \{\psi(F(X_i^t \mid S_i), F(X_j^t \mid S_j)) + (-1)^p\}_{i,j=1}^N$$

Here N stands for the number of instances in the dataset, p=1 when both entities come from the same relation, or from positive examples, and p=0 when entities come form negative examples. Additionally, X_i^t stands for the embeddings of an entity or word, and S_i is the embeddings for the contextual sentence. Moreover, s stands for subject or head entity, and o for object or tail entity.

4 Proposed Method

Our proposed model assumes an embedding model Ω, and a transformation F for obtaining the re-representations from the embeddings. Before giving a detailed description of our proposed method, we characterize it as follows.

1 Given some X_i, as tokens representing sentences, our transformation F obtains a single vector $x_{(out)}^{1d}$ of some dimension d for the joint representation of sentence containing entities e_i^h, and e_i^t, which are related to some relation $R = r_i$.

2 Instead of working with an explicit similarity function such as cosine similarity, our model is trained to maximize the following conditional probability:

$$P_\theta\Big(R = r_i \mid F\big(\Omega(X_i)\big)\Big)$$

With θ being the model parameters.

3 We show that maximizing the above conditional probability as we propose in this section will encourage explicit similarity, as was explained in Section 3.

Given that entities and sentences are sequence of vectors, our transformation F can be such, given a sequence of vectors, it outputs a single vector $x_{(out)}^{1d}$. To that end, our proposed method for F is dynamic routing in capsules. Capsules were introduced first by (Hinton et al., 2011), and dynamic
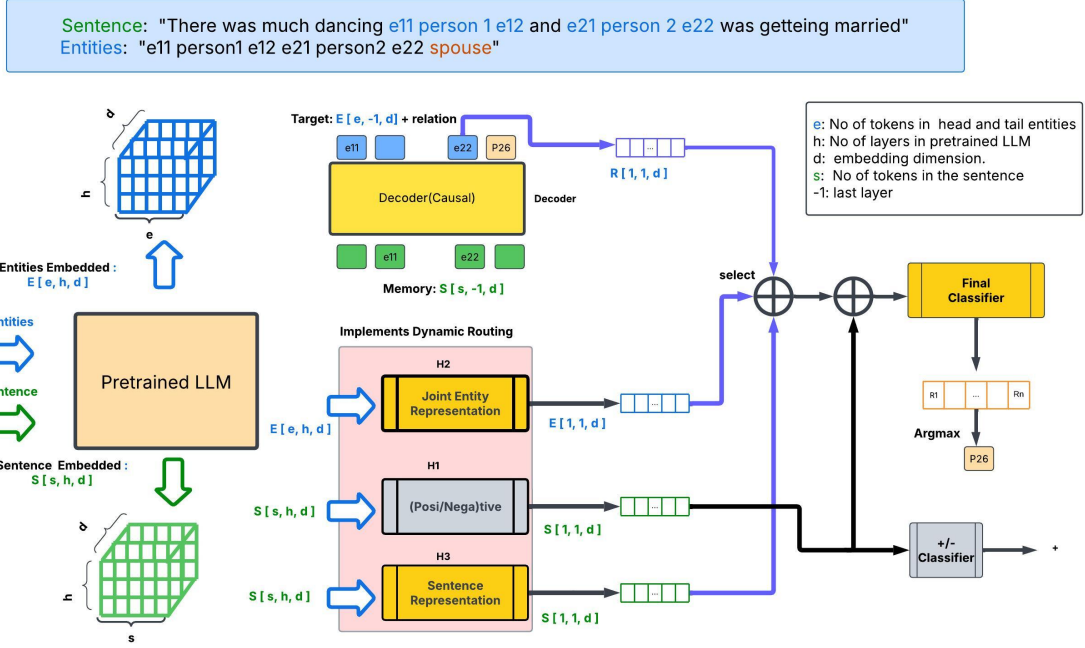
Figure 3: Over all architecture of our proposed model. We use different heads to classify the relation. On top is the Decoder, the head that translate from the sentence to entities post-fixed with relation id. Below it, are heads that implement routing as described in the paper (Heinsen, 2022). In that, H1(gray module in the middle) will identify positivity and negativity of examples. Under it is H3, the head that find the representation of the relation or sentence with entities marked as is shown in text on top of the diagram. Above H1 is H2, the head that calculate the joint representation of concerned entities. To the left is the pre-trained large language model(LLM), the backbone from which we obtain embeddings.

routing in capsules by (Sabour et al., 2017). In capsules network, neurons are grouped into capsules, each capsule representing some particular aspect or feature of the processed entity. Additionally, with dynamic routing, flow of data from a capsule in a layer to a capsule in the next layer depends not only on the weight matrix, but also on coefficients that itself depends on the data, also referred to as routing coefficients. The dynamic routing in capsules network is also called voting by agreement, as a capsule's vote is greater for capsules with which it agrees (Heinsen, 2022).

The dynamic routing algorithm used in this work (Heinsen, 2022), instead of voting by agreement, describe itself as credit assignment system Figure 2.

With that being said, to obtain the embeddings of the given sequence of words or sentence, we use some pre-trained model Ω such as bert_base_uncased (Devlin et al., 2019) or roberta-large (Zhuang et al., 2021):

$$X_{n(inp)}^{hd} = \Omega(X)$$

With X being the tokenized sequence of words for a sentence, and $n_{(inp)}$ being the number of tokens in our input sequence, d being the embedding dimension, and h representing the number of hidden states in the pre-trained model.

To generate an output $x_{(out)}^{1d}$ as representation for the given sentence, the generated multi-dimensional matrix $X_{n(inp)}^{hd}$ is feed into the sequence routing algorithm (Heinsen, 2022). We call the routing module as the routing head. As such, the routing head used, is configured to convert a sequence of vectors of depth h, number of sequence n, representing a sentence, term or entity into a single vector of some dimension d. Additionally, for experiment, we also create some of the routing heads to do some specific predefined tasks, and evaluate if adding them to the main routing head can be of help. To that end, our main routing head is used for obtaining the representation of the sentence with some marking for the concerned entities as is shown in top of the Figure 3. Moreover, as in most datasets, a significant portions of the data are negative, we create an special routing head with two features, one representing positivity (the relation between concerned entities is among our

Dataset	Tacred	Tacredrev	Retacred	Conll04	wikidata
No of Relations	41	41	40	5	353
No of Abstract Entities	23	23	23	4	13533
Train Size	68,124	68,124	58465	1283	372,059
Eval Size	22,631	22,631	19,584	-	-
Test Size	15,509	15,509	13,418	422	360,334
Negative Size	79.5%	79.5%	79.5%	-	29%

Table 1: Statistics of datasets used in this work.

relation set) and another one representing negativity of examples. The working of the mentioned routing head is depicted in the Figure 2. A detailed description of routing heads and the baseline Decoder is depicted in the Figure 3. In practice, we experiment and compare performance of a single routing head, and all routing heads combined with the Decoder, referred here as collection of experts. Each head is characterized as follows:

1. H1: Obtains the representation for positivity or negativity. The output for this head has 2 dimension only, one representing positivity and another negativity Figure 2.

2. H2: This head learns the joint representation of head and tail entities.

3. H3: Is used to obtain the representation for the sentence containing concerned entities. H3 is the main routing head.

4. Decoder: We use a transformer based decoder for the baseline model, as is shown in the Figure 3. As in the example in the mentioned Figure, the decoder uses the last hidden state for the sentence as memory, and entities postfixed by the corresponding relation id as the target.

4.1 Optimization

Given the organization of our data into head and tail entities, e_i^h and e_i^t, and the corresponding sentences S_i for $i \in N$, and relation $r \in R$, with R being the relation set, and the embedding model Ω, and the transformation F based on dynamic routing, and an instance of data as follows:

$$\mathcal{D} = \{(e_i^h, e_i^t, S_i, r_i)\}_{i=1}^N$$

where $r_i \in R$ $\quad and \quad$ $i \in \{1, 2, 3, \dots, N\}$

Where N is the dataset size. The transformation F based on dynamic routing, learns a representation x_i^d, with some dimension d, such that the loss below is minimized:

$$\mathcal{L}(\theta) = -\sum_{i=1}^N \log P_\theta\Big(R = r_i \mid F\big(\Omega(X_i)\big)\Big)$$

Here X_i being the tokens for the sentence S_i, and F is the routing head. In practice, by concatenating the outputs produced by different heads, we experiment if combining heads, also referred to as collection of experts Figure 3 may be of any help. Additionally, if Decoder is among selected heads, its loss will be added to the classifiers loss as in the Figure 3.

5 Experiments

5.1 Datasets

We test our model on several sentential relation extraction datasets. Specifically we test the proposed model on wikidata (Sorokin and Gurevych, 2017), Tacred (Zhang et al., 2017), Tacredrev (Alt et al., 2020), Retacred (Stoica et al., 2021), and Conll04 (Roth and Yih, 2004; Markus and Adrian, 2020). In all datasets, except Conll04, negative example make a significant portion of the examples. To that end, there are several factors to note about the datasets.

1. The ratio of positive and negative examples: Conll04 has no negative record, while wikidata has 22/29% of example as negative, and all Tacred variants have 79.5% of example as negative.

2. Number of entity types or manual abstract label of entities: All Tacred variants has 23 abstract labels for entities as and according to name entity recognition types in stanford

NER system (Zhang et al., 2017). Conll04 has only 4 different types of entities, and Wikidata has the highest number of abstract labels for entities 13,533.

3. Number of relations: From number of relation points of view, wikidata has 353 relation types, which is the largest among all datasets considered, while Conll04 has only 5 types of relations.

5.2 Different Configuration of the Sentence

The assumption for our proposed model is that, in order to do re-representation, dynamic routing can do feature selection . As such, we study our proposed model with different settings or configuration of the sentence. Accordingly, a given sentence can provide different level of details about the entities and their relations. As example, the sentence "<Mask> was getting married to <Mask>.", wherein the two concerned entities are masked, provide less details as when entities are not masked. Similarly, when entities are replaced by entity type, the level of details are less than the original sentence. This is important as in order to do re-representation, models need to do abstraction, and discard some unnecessary details. As such, manual abstract label of entities or entity types can perhaps make the job of RE models on some datasets easier. To that end, the following sentence configurations are used with markings as is shown in the Figure 3:

- Abstract: We replace surface form of the entity with entity type (abstract label of the entity). example: Germany or France is replaced by entity type Country.

- Mask: We replace surface form of the entity with the placeholder, 'MASK', in the sentence.

- Entities: We use only surface form of the entity as is.

- Mix: The entity type, or abstract label for entity and its surface form is used together with some marking. Example: "x was getting married to y." is transformed into : " [e11] + person * x [e12] was getting married to [e21] # person & y [e22]."

Config	Model	Retacred	Conll04
Mix	H3	**92.2(80.1)**	**100.0(100.0)**
	Decoder	49.3(21.0)	78.6(79.8)
Entities	H3	**89.7(58.5)**	**84.1(84.7)**
	Decoder	50.4(31.5)	42.1(41.8)
MASK	H3	**81.7(54.2)**	**80.1(79.3)**
	Decoder	–	–
Abstract	H3	**75.2(48.5)**	**82.2(80.3)**
	Decoder	29.1(13.0)	61.8(63.7)

Table 2: Comparative performance of the routing head H3, and transformer based Decoder on different configuration of sentence or information granularity. Recorded scores inside parenthesis are F1 Macro, and F1 Micro otherwise. The backbone model is roberta-large.

5.3 Experiment One: Comparative Performance on Different Information Granularity

We investigate performance achievable with our proposed model, and the transformer based Decoder on each sentence configuration described above. As each configuration of the sentence provide different level of details about entities and their relation, we refer to different sentence configuration as different information granularity. Accordingly, the relation between two entities in a sentence can be mostly predicted in all sentence configuration considered here; However, the best result by the proposed model is when the entity type is added to contextual sentence. For Decoder however, the best result changes across datasets considered Table 2. As such, on Retacred, Decoder's best score is when entity type is not added to the sentence(configuration "Entities"). However on Conll04, it is the other way around. Moreover, on Retraced, Decoder have relatively low scores, while on Conll04, our Decoder's score(78.6 F1 Micro) is above state of the art (with state of the art being 76.5, Tables 5, and Table 2).

5.4 Experiment Two: Entity Types as Manual Label for Re-Representation

Given that entity types increase the similarity as is expected for re-representation (a depicted example can be seen in the Figure 1), we can view entity types as manual label for re-representation. To that end, we extract entities from the respective sentences, and train the proposed model on the extracted entities and entity types. This help

us study entities and entity types in isolation. In doing so, we consider all sentence configurations (except Mask) as explained in the Section 5.2. As such, for Conll04, our proposed model exhibit same performance with configuration Abstract and configuration Mix Table 4. As such, we can conclude that, on some datasets, the manual label for re-representations or entity types (configuration Abstract), result in best performance. A possible explanation would be: when the entity types or manual label of re-representation can predict the relation, or given a relation the entity types for head and tail entities can be predicted, such as in Conll04, entity types alone (config Abstract) result in peak performance and less complexity Table 4.

5.5 Experiment Three: Performance on Varying Number of Entity Types

Does increasing the number of relation and entity types, or increased complexity for re-representation effect performance? To that end, we already observed the relatively good performance by Decoder on Conll04, the dataset with 4 entity types and 5 relations only Table 2. As such, we also evaluated the Decoder and the proposed model H3, on the smaller subset of Retacred, person-person, having only 6 relations and 1 entity type only. Additionally, after training on the full dataset, we recorded the performance on the same subset, person-person*. Accordingly, Decoder's performance is better when number of relation and entity types are smaller Table 3, as was noted for Conll04. As such, the experiment support the notion that transformer based Decoder changes performance across dataset presumably due to larger number of entity types, and relations. Unlike the Decoder, the proposed model exhibit relatively high performance across datasets, with different number of relation, and entity types.

Dataset Subset	H3	Decoder
Full	92.2(80.1)	49.3(21.0)
Person-Person	93.0(82.2)	72.6(60.4)
Person-Person*	89.7(78.3)	51.6(38.2)

Table 3: Performance on varying number of relation and entity types (config mix).Values inside the parenthesis are F1 Macro, and F1 micro otherwise. Person-Person is the subset of Retacred having head and tail entity types as person only. It is the largest subset of Retacred categorized by head-tail entity types. Full is the entire dataset. Person-Person* is performance on the same subset, but by the model trained on the full Retacred.

Metrics	Retacred	Conll04
Mix	**71.7**	**100.0**
Entities	71.3	48.1
Abstract	62.0	**100.0**

Table 4: RE using entities extracted from the sentence, and with routing head H2. The backbone model here is roberta-large. The reported values are F1 micro.

5.6 Experiment Four: Comparison with State of the Art

To compare with state of the art, we trained our proposed model on the mentioned datasets, and documented the result. The result is shown in the Table 5. Our observations show that our proposed model outperforms state of the art on 4 datasets. To that end, our routing head H3, with roberta-large as the backbone, keeps a relatively high performance on all datasets. It outperform state of the art on all dataset, except Wikidata. In the the Section 6.2 we show that noise is the main reason for the low performance on Wikidata. Moreover, despite the extra complexity that use of all heads, or expert heads, adds to our main model, we noticed little improvement. We therefore did not evaluate the expert head on Wikidata. Lastly, for our proposed model H3, the difference with bert-base-uncased (Devlin et al., 2019) and roberta-large (Zhuang et al., 2021) as the backbone is noticeable.

6 Observations

6.1 Re-Representation in Neural Network

As suggested initially, treating sentential RE as analogy, requires some form of re-representation to improve the match. To check if neural-network also does re-representation, using a subset from Retacred test set, we create positive and negative analogous entities of the form $e_i^h : e_i^t :: e_j^h : e_j^t$, for all $i, j \in \{1, 2, .., N\}$ such that the corresponding sentence S_i and S_j expresses the same relation between the corresponding entities in positive examples and different relation in negative examples. In doing so, we obtain the embedding for a given entity in the sentence, by feeding the sentence into the backbone model, and then slice the entity from the sentence embedding. We then calculate the cosine similarity, and pairwise euclidean distance between respective head, and tail entities in both positive and negative examples. We calculate the mentioned values across hidden states
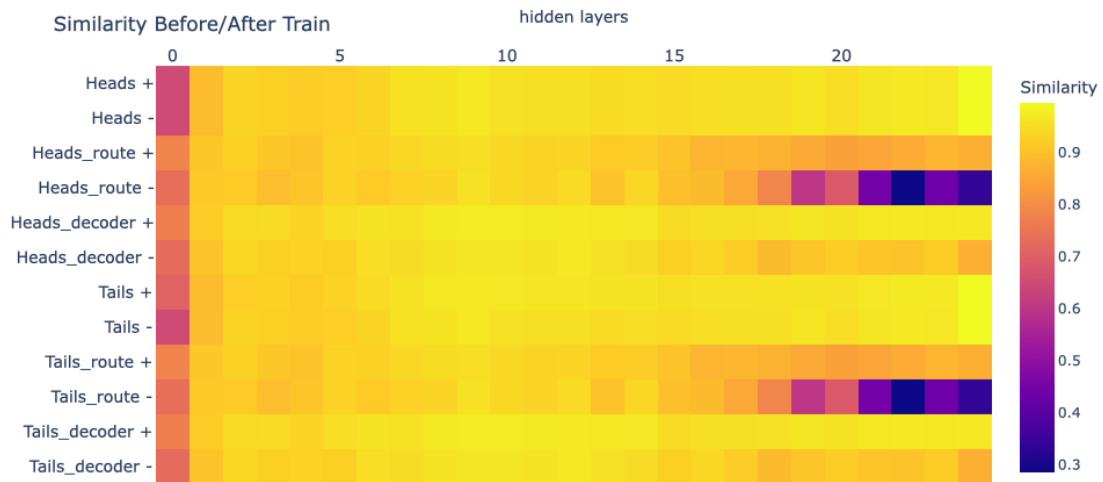
(a) As can be seen, before training, the similarity between head and tail terms in positive (Heads +, Tails +) and negative (Heads -, Tails -) examples are barely distinguishable. However, after training, the model based on dynamic routing, does a good job of making head/tail terms more similar in positive examples, and dissimilar in negative examples.
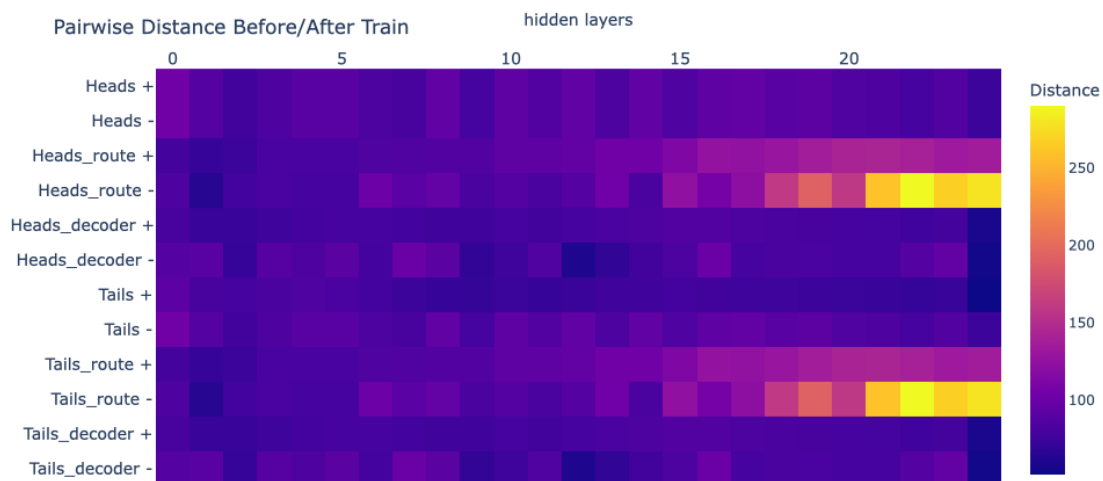
(b) The distinction between positive and negative examples are barely distinguishable before training both for head terms (Heads +, and Heads -) and also for tail terms(Tails +, Tails -). However, after training, and that also specially for the model based on dynamic routing (*_route +, *_route -) the increase in the distance between head/tail terms in positive examples, are far less intense than in negative examples.

Figure 4: X-axis represent different hidden layers of the pre-trained LLM. Y-axis represent categories for which representation's similarity or distance was calculated. + represent positive analogous examples, and - represent negative analogous examples respectively. Heads and Tails are the related head and tail terms in proportional analogy. As example in king:queen::man:woman , (king, man) are head, whereas (queen, woman) are tail. We report the result of calculations obtained on representation after training with routing heads H3(Heads/Tails_route +/-) , and transformer based Decoder (Heads/Tails_decoder -/+). We also report the same before training (Heads/Tails +/-).

Model	Tacred	Tacredrev	Retacred	Conll04	Wikidata
Entity Marker (2022)	74.6	83.2	91.1	-	-
Curriculum Learning(2021)	75.2	-	91.4	-	-
REBEL (2021)	-	–	90.4	76.5	-
KGpool (2021)	-	-	-	-	**88.6**
RAG4RE (2024)	86.6	88.3	73.3	-	**-**
Ours bert $_{H3}$	84.8 (47.8)	85.3 (49.7)	89.4 (74.0)	99.7(99.8)	84.5 (32.0)
Ours bert $_{H1,H2,H3,Decoder}$	**87.4** (48.3)	88.7(50.9)	88.7 (68.5)	**100.(100.)**	–
Ours Roberta $_{H3}$	87.1 **(61.1)**	**88.8 (64.2)**	**92.2(80.1)**	**100. (100.)**	85.6 (32.9)

Table 5: Our method's performance compared with state of the art. Best score is bold, state of the art is blue. Values inside the parenthesis are F1 Macro, and F1 micro otherwise. The configuration of sentence is Mix, and backbone is as indicated. We do not test all heads(H1,H2,H3,Decoder) for Wikidata as we found H3's performance to be already good on Wikidata's noisy labels.

of pre-trained backbone model, both after training with each training heads H3 and Decoder, and also before training, and then create a heat map as is shown in the Figure 4a. Accordingly, "Head +" , and "Head -" represent the cosine similarity between heads in positive and negative examples before training. As can be seen, the similarity is not much different between positive and negative examples. However, for the proposed model, after training, the similarity decreases significantly in negative examples, making the difference between positive and negative examples clearly noticeable (specially in final layers of the backbone model). Similarly, the pairwise euclidean distance between positive and negative examples, shown in Figure 4b, after training are clearly distinguishable for the proposed model(Heads_route +, Head_route -) than it is for the vanilla Decoder (Heads_decoder +, Head_decoder -).

6.2 Noise in Wikidata's Labels

The tow Tacred variants (Tacredrev, Retacred) are very good attempts to improve data quality and reduce error rate in the Tacred. Each of these datasets improve model performance with 8.0% and 14.3% F1 score over the original Tacred respectively. In comparison to Tacred, wikidata has much larger and diverse types of relations. Its quality however has not gone a similar study. Instead, a significant attention has been given in improving model performance by incorporating extra additional details through complex models. As our model's performance is below state of the art on Wikidata, we were intrigued to have a look at examples in which our model disagree with labels from the dataset.

Not surprisingly though, we found out that a significant portion of errors are due to confusion in the dataset labels. As example, for instances which our model disagree with the dataset label, the labels seem random. More such examples, and statistics in Appendix B.2. We categorize all examples that our model disagree with labels from dataset in the appendix B.2, Table 6 , Table 7, and Table 8.

7 Limitations

Over all the dynamic routing proposed by (Heinsen, 2022) is efficient and scalable as is explained in the original paper. However using all routing heads as collection of experts increases the complexity n folds, where n is the number of routing heads in the model. However, the good news is that, perhaps a single H3 head can do a better job as is shown in the Table 5.

8 Conclusion and Future Research Directions

In this paper we improve sentential relation extraction performance on several benchmarks. Additionally we identify noise as one of the main cause for low performance on largest sentential RE benchmark Wikidata. Furthermore, we propose re-representation as one of the challenges of sentential RE models. Lastly, we show that sentential RE dataset may not be as much sentence dependent as expected B.1. For future research direction, we are planing to study word analogies of the form a:b::c:d, jointly with sentential RE datasets. Specifically, it would be interesting to see how much improvement can training sentential RE benchmarks bring to word analogy benchmarks.

Ethics Statement

We did pay for the dataset Tacred. Other datasets are publicly available. In addition for the proposed algorithm being efficient, we tried to minimize our CO_2 footprint. As such, this work comply and adhere to ACL code of ethics.[2]

References

Christoph Alt, Aleksandra Gabryszak, and Leonhard Hennig. 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1558–1569, Online. Association for Computational Linguistics.

Anson Bastos, Abhishek Nadgeri, Kuldeep Singh, Isaiah Onando Mulang, Saeedeh Shekarpour, Johannes Hoffart, and Manohar Kaul. 2021. Recon: Relation extraction using knowledge graph context in a graph neural network. In *Proceedings of the Web Conference 2021*, WWW '21, page 1673–1685, New York, NY, USA. Association for Computing Machinery.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. Preprint, arXiv:1810.04805.

Sefika Efeoglu and Adrian Paschke. 2024. Retrieval-augmented generation-based relation extraction. Preprint, arXiv:2404.13397.

Dedre Gentner. 1983. Structure-mapping: A theoretical framework for analogy. *Cognitive Science*, 7(2):155–170.

Dedre Gentner and Kenneth Kurtz. 2006. Relations, objects, and the composition of analogies. *Cognitive science*, 30:609–42.

Franz A. Heinsen. 2022. An algorithm for routing vectors in sequences. Preprint, arXiv:2211.11754.

Geoffrey E. Hinton, Alex Krizhevsky, and Sida D. Wang. 2011. Transforming auto-encoders. In *Artificial Neural Networks and Machine Learning – ICANN 2011*, pages 44–51, Berlin, Heidelberg. Springer Berlin Heidelberg.

Pere-Lluís Huguet Cabot and Roberto Navigli. 2021. REBEL: Relation extraction by end-to-end language generation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2370–2381, Punta Cana, Dominican Republic. Association for Computational Linguistics.

John E. Hummel and Leonidas A. A. Doumas. 2023. Analogy and similarity. In Ron Sun, editor, *The Cambridge Handbook of Computational Cognitive Sciences*, 2 edition, Cambridge Handbooks in Psychology, pages 451–473. Cambridge University Press.

Jean-Daniel Zucker Lorenza Saitta. 2013. *Abstraction in Artificial Intelligence and Complex Systems*. Springer, New York, NY.

Eberts Markus and Ulges Adrian. 2020. *Span-Based Joint Entity and Relation Extraction with Transformer Pre-Training*. IOS Press.

Abhishek Nadgeri, Anson Bastos, Kuldeep Singh, Isaiah Onando Mulang', Johannes Hoffart, Saeedeh Shekarpour, and Vijay Saraswat. 2021. KGPool: Dynamic knowledge graph context selection for relation extraction. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 535–548, Online. Association for Computational Linguistics.

Seongsik Park and Harksoo Kim. 2021. Improving sentence-level relation extraction through curriculum learning. Preprint, arXiv:2107.09332.

Sebastian Riedel, Limin Yao, and Andrew McCallum. 2010. Modeling relations and their mentions without labeled text. In *ECML/PKDD*.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004*, pages 1–8, Boston, Massachusetts, USA. Association for Computational Linguistics.

Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. 2017. Dynamic routing between capsules. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 3859–3869, Red Hook, NY, USA. Curran Associates Inc.

Daniel C. Silliman and Kenneth J. Kurtz. 2019. Evidence of analogical re-representation from a change detection task. *Cognition*, 190:128–136.

Daniil Sorokin and Iryna Gurevych. 2017. Context-aware representations for knowledge base relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1784–1789, Copenhagen, Denmark. Association for Computational Linguistics.

George Stoica, Emmanouil Antonios Platanios, and Barnabás Póczos. 2021. Re-tacred: Addressing shortcomings of the TACRED dataset. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13843–13850. AAAI Press.

---

[2]https://www.aclweb.org/portal/content/acl-code-ethics

Shikhar Vashishth, Rishabh Joshi, Sai Suman Prayaga, Chiranjib Bhattacharyya, and Partha Talukdar. 2018. RESIDE: Improving distantly-supervised neural relation extraction using side information. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1257–1266, Brussels, Belgium. Association for Computational Linguistics.

Yuhao Zhang, Victor Zhong, Danqi Chen, Gabor Angeli, and Christopher D. Manning. 2017. Position-aware attention and supervised data improve slot filling. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing (EMNLP 2017)*, pages 35–45.

Wenxuan Zhou and Muhao Chen. 2022. An improved baseline for sentence-level relation extraction. In *Proceedings of the 2nd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 12th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 161–168, Online only. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A robustly optimized BERT pre-training approach with post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

A Training and reproducibility

For all routing heads we use the code from (Heinsen, 2022). Additionally, we use tokenizers from https://huggingface.co for bert-base-uncased, and roberta-large respectively. Our backbone models are too from https://huggingface.co. Furthermore, for routing head H3, we train it on datasets Tacred, Retacred, and Tacredrev with batch size 64, learning rate 10^{-5} , and on the dataset Wikidata with batch size 128, and similar learning rate as for Tacred and its variants. For collection of experts we use an smaller batch size of 24. For optimizer with use Adam from torch.optim. Moreover, we find hidden state of routing heads to have great influence on performance. To that end, for H3, w used hidden_d=256, and out_dimension=512. Moreover, we trained the proposed model for Tacred and its variants for 6 epochs, while we trained only for 1 epoch on Wikidata.

Another point to note is: In case of wikidata, when entities did not have an entity type (or instance of as in the dataset), we checked the Wikidata knowledge base to retrieve parent class as entity type[3] . Furthermore, when entities had several values as "instance of" or parent class, again we query Wikidata to check if they have a common parent class, and used the parent class as the entity type, if not, the most common class was uses for entity type. Lastly, unless explicitly mentioned, all experiments are done with Roberta-Large as backbone.

B Observation

B.1 Are Sentential RE Datasets Truly Sentential?

To answer if relation between the entities, can be inferred without reading the sentence, and only be looking into entities, we trained and evaluated the proposed model on entities with configuration as was discussed for the sentence. The result for different configurations are recorded in the Table 4. Accordingly, most relation can be inferred without reading the concerned sentences.

B.2 Noise in Wikidata's Labels

On examples which our model disagree with the dataset Wikidata, we found a pattern. Specifically, given a pair (p0-p*), where p0 is label('no relation') provided by the dataset, and p*(some relation other

than "no relation") predicted label, there is usually another category of predictions as (p*-p0). In both groups of examples, the probability that p * is true is similar, regardless of the label provided by the dataset. The group pairs, such as p0-p17 and p17-p0; show confusion caused as a result of incorrect labels. Some examples in Table 6.

For ease of understanding, we list Wikidata relation codes used in the table with corresponding labels as: • P131(located in the administrative territorial entity) • P17(country) • P47(shares border with) • P118(league) • P571(inception) • P47(shares border with) • P361(part of) • P463(member of)

Label-Prediction	probability of P* being True	count
P0-P17	80.6	4090
P0-P131	90.0	4037
P0-P47	60.0	3518
P0-P118	70.0	2155
P0-P571	50.0	1718
P0-All	70.0	29021
P47-P0	60.0	12184
P131-P0	80.0	4775
P17-P0	70.0	4312
P361-P0	60.0	2152
P463-P0	70.0	1546
All-P0	60.0	40155
label!=prediction	-	106534

Table 6: Top categories(sorted) on which model's predictions does not match with the label from benchmark. * represent a relation other than 'no relation'. The probability here is calculated by sampling 10 random example from each category, and then manually checking if p* holds.

---

[3]https://query.wikidata.org

Label-Prediction	Sample
P0-P17 P0:No relation P17:country	Los Dominicos is a metro station on Line 1 of the Santiago Metro in Santiago , Chile, and is also the eastern terminal of this line .
P0-P131 P0:No relation P131: Located in the administrative territorial entity	Boechout is a railway station in Boechout , Antwerp , Belgium .
P0-P47 P0:No relation P47:Shares border with	There are now approximately twenty restaurants in operation in Georgia , and about nine more in North Carolina , South Carolina , Florida , and Tennessee.

Table 7: Random sample from $P0 - P^*$, where p^* is any relation from relation set other than no relation, and p0 is no relation

Label-Prediction	Sample
P6-P138 P6:head of government P138:named after	She later served in the Blair ministry under Prime Minister Tony Blair in a number of roles , becoming Britains first female Foreign Secretary in 2006 .
P264-P136 P264:record label P136:Genre	CD1 is the unofficial name of an untitled album by English industrial music band Throbbing Gristle , released in 1986 through record label Mute .
P1416-P102 P1416:affiliation P102:member of political party	Other famous Solidarity activists such as [e11] Anna Walentynowicz Solidarity activists such as Anna Walentynowicz , Zbigniew Romaszewski and Antoni Macierewicz have visited the Basilica as well .

Table 8: Random sample from $p^* - p^*$, where p^* is any relation from relation set other than no relation. Consider the first row in which both label and prediction is correct.