One-shot relation retrieval in news archives: adapting N-way K-shot relation classification for efficient knowledge extraction

Hugo Thomas, Guillaume Gravier, Pascale Sébillot

▶ **To cite this version:**

Hugo Thomas, Guillaume Gravier, Pascale Sébillot. One-shot relation retrieval in news archives: adapting N-way K-shot relation classification for efficient knowledge extraction. KES 2024 - 28th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems, Sep 2024, Seville, Spain. pp.1-10. hal-04708239

HAL Id: hal-04708239
https://hal.science/hal-04708239v1

Submitted on 24 Sep 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2024) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

28th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2024)

One-shot relation retrieval in news archives: adapting N-way K-shot relation classification for efficient knowledge extraction

Hugo Thomas, Guillaume Gravier, Pascale Sébillot

Univ Rennes, CNRS, Inria, IRISA - UMR 6074, France

Abstract

One-shot relation retrieval is the knowledge extraction task that consists in searching in a textual dataset for all occurrences of a relation of interest, named the source relation, characterized by a single example—a relation being a link between a pair of entities in an utterance. Performing this task on large datasets requires an intelligent system to automate the process, for instance when exploring news archives for press review or business intelligence. We propose a framework that leverages the representation learning capabilities of N-way K-shot models for few-shot relation classification and extends these models to enable one-shot retrieval with a rejection class. At evaluation time, one-shot relation retrieval is performed in a N-way K-shot setting where 1 of the N ways (or relations) is the source relation and the N-1 others are distractors, i.e., relations modeling a rejection class. We benchmark this framework and investigate the influence of the number and the choice of distractors on the standard TACREV and FewRel datasets. Experimental results demonstrate the effectiveness of our approach to address this highly challenging task, however with high variability primarily induced by the type of the source relation. Experiments also highlight a sound strategy for the choice of distractors—a large number of distractors at an intermediate distance from the embedding of the source relation in the latent space learned by the model—, which provides a competing trade-off between recall and precision. This strategy is globally optimal but can however be surpassed on certain source relations by others, depending on the characteristics of the source relation, paving the way for future work. We finally show the substantial benefit of two-shot retrieval over one-shot retrieval, which sheds light on the design of actual intelligent applications leveraging one- or few-shot relation retrieval.

© 2024 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the KES International.

Keywords: relation extraction; relation retrieval; few-shot learning; N-way K-shot learning

1. Introduction

Knowledge acquisition and intelligence gathering from large news archives meet various application domains in media technology and intelligence business. Focusing on the textual nature of these archives, a key task for such applications is that of relation extraction which consists in finding all occurrences of a given relation between two entities in the corpus. This task is often addressed in a knowledge base population setting with a set of pre-defined relations to detect (see, e.g., the survey of Zhong et al. [30]). For instance, in business intelligence, one would consider detecting all utterances in the daily press that refer to any company A buying any other company B to add this fact as

1877-0509 © 2024 The Authors. Published by Elsevier B.V.
This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/4.0/)
Peer-review under responsibility of the scientific committee of the KES International.

a triplet into a knowledge base. Regardless of named entity detection and disambiguation, most approaches assume the set of possible relations to be known in advance, thus enabling to train relation detection and classification models once and for all.

In this paper, we adopt a different approach towards an intelligent and interactive system where a user selects an occurrence of a relation, designated as the source relation, and asks the system to retrieve all (other) occurrences bearing the same relation in the corpus from this single example. Note that "same relation" denotes a relation of the same (unknown) type as the source one (e.g., "buying" in the previous example) and the entities being related may differ from the source. Journalists are typically interested in such an application to feed their article with related facts or find a new interesting angle. Investigative journalism is also in dire need of such an application when mining large amounts of documents such as the Panama papers in search of new facts that are hard to foresee. Compared to the standard relation extraction approach, this one-shot retrieval task adds a few difficulties, at the forefront of which the fact that the relation of interest (or source relation) is defined solely through a single example, lacking typing or further characterization to help the definition of an efficient model.

Among the vast literature on relation detection and extraction [7, 29, 20], (lexico-)syntactic pattern based and semi-supervised (especially bootstrapping) techniques [16, 25, 6, 13] offer a first possible approach to address this relation retrieval task. However, they can hardly be used in our scenario, where the only example at disposal hinders the building of a reliable seed and/or retrieval pattern. Unsupervised techniques such as open information extraction ones [21], which aim to extract all possible relational triplets from a large corpus potentially under the control of (loose) syntactic and lexical constraints, are also too far from our objective and cannot be sufficiently driven by a single source relation example. Alternatively, transformer-based large language models (LLMs) are known to achieve state-of-the-art performance in relation classification with a predetermined set of classes, provided large amounts of training data [7, 29]. This is typically addressed by fine-tuning LLMs trained on a huge amount of data to capture rich information on language that benefits downstream tasks. Fine-tuning however requires enough training data, not available in a one-shot relation retrieval scenario, and classes to be defined beforehand. To take advantage of the potential of LLMs, we thus focus on a few-shot learning paradigm known as N-way K-shot. This paradigm has been successfully used for few-shot relation classification [14, 22, 24, 10, 4, 5, 18], obtaining state-of-the-art performance using transformer encoders in a data-scarce relation classification context. In the N-way K-shot paradigm, each batch contains N classes (among $M \geq N$ possible ones) with K samples each (K is usually small) and a number of samples to be classified as one of the N classes present in the batch. Although N-way K-shot allows to perform few-shot relation classification, it requires adaptation to suit our one-shot relation retrieval requirements. In particular, contrary to relation classification, the retrieval task only comes with a single possible class that is either present or not in an utterance. Moreover, this class and/or its support utterance might frequently change depending on the user's interest. Retraining one of the aforementioned classification model is thus impractical.

In this article, we propose and extensively study RaReMUD—Rare Relation Mining Using Distractors—an approach to turn N-way K-shot relation classification into a retrieval model that suits our intelligent system use case. We address in particular two key differences with standard N-way K-shot learning. On the one hand, the training and evaluation (or operational) phases feature distinct tasks, which is not the case in few-shot classification: here, the training phase is used for representation learning, and leveraged in the operational phase for detection, comparing relations via the learned representations. On the other hand, and more fundamentally, retrieval requires a rejection mechanism to decide that the relation to retrieve is not present in an utterance. In our approach, the rejection mechanism implies the selection of competing relations for a given source relation, designated as distractors, thus enabling to form batches with N classes—the one of interest complemented with N-1 distractors.

The paper first describes the task and methodology at hand, the paradigms of N-way K-shot and prototype learning, as well as the architecture and specifics of RaReMUD. The efficiency of the latter is evaluated on the TACREV [1] and FewRel [15] datasets to demonstrate its usability in real-world applications. We then experimentally study the impact of the number of distractors, a key parameter of RareMUD, and compare strategies to select the distractors for a given source relation. Experiments also study the influence of the number of examples for the source relation on RaReMUD's score to assess the difficulty induced by the one-shot aspect of the task. Finally, our post-hoc analysis of optimal distractors given a source relation reveals the room for improvement that RaReMUD could achieve using a better, yet unknown, distractor choice strategy.

2. Related Works

To the best of our knowledge, one-shot relation retrieval has not been explored in the literature. Its objective of detecting all occurrences of a specific source relation in a text corpus is, however, shared by works that rely on semi-supervised learning [7, 29, 20]. Distant supervision (e.g., [25]) and bootstrapping methods have been proposed, the latter requiring a seed formed either by some pairs of entities known to be linked by the source relation (e.g., [16, 3, 2]), or by some lexico-syntactic patterns of the relation (e.g., [6] or the BRET variant of Gupta et al. [13]). These seeds allow to iteratively find new pairs of entities in utterances of the corpus that match the patterns already known, or abstract the contexts in which they appear to obtain new patterns, until a stopping criterion is satisfied. However, the seeds in these works contain more than just the single example of the source relation of our use case, sole example which appears too poor to extract a reliable pattern to bootstrap the iterative process.

Concerning recent supervised learning methods, LLMs have shown their ability to achieve state-of-the art performance in the task of relation classification with a predetermined set of classes and when large amounts of training data are available [7]. Note that conversational models have also recently been used as generative models with prompt engineering for few- or zero-shot relation extraction [12, 28, 9]. While prompt engineering is appealing, it is not particularly suited for one-shot retrieval as it can hardly scan a large set of utterances to verify whether they express the same relation as the source one or not. Supervised deep learning-based methods close to our task typically address one-shot and few-shot relation classification, differing from our task in at least two important aspects: relation retrieval is a detection task, not a classification one; relation classification is often performed as a closed-set task, i.e., with a fixed number of existing identified relation types, whereas relation retrieval is an open-set task, with a virtually unlimited amount of relations to detect and of unidentified relations to reject. Recent examples of such methods are transformer-based models trained and evaluated with the N-way K-shot paradigm [14, 22, 24, 10, 4, 5, 18]. Interestingly for our work, these models have recently been granted the ability to reject a sample as belonging to one of the N (from N-way K-shot) relations of a batch, using a rejection vector called NOTA—None Of The Above—vector [11]. In this work, the NOTA vector is added as a representation of a rejection class to each batch, in addition to the N relations. Contrary to the relations, whose respective representations are derived (or learned in training) from the K corresponding samples, the NOTA vector has no supporting utterance and is directly learned as a prototype for rejection. This idea was later extended in [23], adding several rejection vectors to represent the rejection class more effectively. RaReMUD borrows from the idea of multiple rejection vectors to represent the absence of the relation of interest, exploiting instead actual relations, called distractors, with their support utterances to yield rejection prototypes. This decision is induced by the open-set nature of the task which calls for rejection vectors specific to the source relation, preventing to learn fixed rejection prototypes.

3. Task and Methodology

We first formalize the task of one-shot relation retrieval, introduce the notations of the N-way K-shot prototype learning paradigm before detailing the RaReMUD framework and its specifics.

3.1. Task definition

In utterance-level relation classification as well as relation retrieval, we consider quadruples of the form (s, h, r, t), where utterance s supports a relation $r \in \mathcal{T}$ between the head and tail entities $(h, t) \in \mathcal{E}^2$, $\mathcal{T}$ and $\mathcal{E}$ respectively being the set of all possible relation types and the set of all entities. Entities can be named entities or pronouns referring them. For instance, in the sentence s "Sucre is the capital city of Bolivia.", the relation between h = "Sucre" and t = "Bolivia" is of type r = "capital_city_of". We consider only binary relation types and admit that a unique relation exists between two entities in a given utterance. Note however that an utterance can support up to twice as many quadruples as there are pairs of entities in it, considering that most relations are not symmetrical. In the relation classification task, labeled training data therefore consists of quadruples (s, h, r, t) to learn models that predict the relation $r \in \mathcal{T}$ between a designated pair of entities in an utterance s. For this task, generalization capabilities are often measured by training on a relation set $\mathcal{T}_b$ found in a dataset split $\mathcal{D}_b$, evaluating on a distinct dataset split $\mathcal{D}_n$ with a distinct relation set $\mathcal{T}_n$ ($\mathcal{T}_b \cap \mathcal{T}_n = \emptyset$). In the one-shot relation retrieval task, a single annotated quadruple

(s, h, r, t) is given and the task consists in finding all utterances along with the pair of entities therein that support the source relation r. This definition thus calls for few-shot learning paradigms to take advantage of pre-trained LLMs in a one-shot context.

3.2. N-way K-shot prototype learning for relation classification

In few-shot learning, the N-way K-shot paradigm is often used to simulate training and evaluation from few examples, relying on prototype learning to represent each of the classes with a unique vector derived from the few examples available. In all generalities, the N-way K-shot paradigm consists in making batch-based decisions with a specific construction of batches. Each batch consists of a support set with N distinct relation types (or classes) randomly picked among all available ones, each with K support examples, noted $\mathcal{S} = \{s_k^i; i = 1, \ldots, N; k = 1, \ldots, K\}$. Every batch also includes a so-called query set of L examples, noted $\mathcal{Q} = \{s_j; j = 1, \ldots, L\}$, to be classified as one of the N classes present in the support set $\mathcal{S}$. Classification of utterances in $\mathcal{Q}$ relies on the construction of a prototype for each of the N relation types in $\mathcal{S}$, prototype obtained from the embeddings of the corresponding support utterances. In other words, the relation supported by utterance s_k^i is embedded into a fixed size vector e_k^i to yield a prototype e^i of relation i in the batch by combining $e_1^i, \ldots, e_K^i$. An utterance s_j of the query set is classified by comparing the embedding of the relation supported by s_j with each prototype given a distance metric. The corresponding model is thus defined through the three main following components:

1. relation embedding: this corresponds to the model used to embed the relation supported by an utterance where the entities of interest are marked, which is where pre-trained LLMs are handy;
2. prototype construction: this specifies how the prototype for a relation is constructed from the K embeddings of its support utterances;
3. prototype comparison: this last component specifies the metric used to compare prototypes and relation embeddings, typically relying on either a standard distance, such as cosine or Euclidean, or metric learning techniques.

As stated in Section 3.1, such models are often trained and evaluated on two distinct sets of relations to assess the model in an actual K shot scenario. Indeed, a relation present in the training set in fact benefits from more than K examples in total to train the model's components as it might appear in several batches. Hence the need to use a different set of relations at evaluation time, where parameters no longer vary, to ensure proper few-shot learning.

3.3. RaReMUD, an N-way K-shot approach for one-shot relation retrieval

While efficient for few-shot relation classification, the previous N-way K-shot paradigm does not apply straightforwardly to our use case of one-shot relation retrieval. We thus introduce a framework called RaReMUD, exploiting the N-way K-shot batching strategy to perform detection as follows. First, an N-way K-shot classification model is trained in a standard relation classification manner with very few shots on a large database of known relations in order to learn the prototype representation and comparison metric components of the model. During the operational retrieval phase for a given source relation, we exploit a particular batch structure for retrieval. In each batch, the query set $\mathcal{Q}$ consists in a collection of L utterances where the presence of the source relation is to be detected. To enable retrieval, the support set $\mathcal{S}$ is defined as follows: One of the N-way is naturally the source relation to detect in utterances of $\mathcal{Q}$, characterized by a single utterance. The key idea of RaReMUD is to complement the source relation with $N - 1$ other relations, called *distractors*, modeling the rejection class, thus casting the detection problem of retrieval as an N-way K-shot relation classification task. N relation prototypes are generated—one representing the source relation, the others the distractors—using K=1 utterance for the source relation and the same K value as used during training for the distractors. Each utterance in the query set is compared to the N prototypes and deemed as supporting the source relation if closer to its prototype than to any of the distractors' prototypes, or rejected otherwise.

The number (i.e., choosing N at retrieval time) and choice of the distractors with respect to a given source relation is naturally key: we propose below various strategies for the choice of distractors that we experimentally compare in this paper. It is also important to note that the training phase focuses on learning the representation and metric components to use for retrieval. It therefore does not require K=1, contrary to what is done in the retrieval phase for the source relation. Similarly, the number of ways can differ between training and retrieval.

Fully specifying a RaReMUD model thus requires defining (a) the specific architecture of the different components and (b) a strategy to choose the distractors, which we detail below.

3.3.1. Architectures

In all experiments, relation encoding relies on a LLM, following [26], fine-tuned in the training phase. The entities of interest in an utterance are surrounded by tokens <E1> and </E1> for the head and <E2> and </E2> for the tail. The corresponding relation embedding is obtained by concatenating the embeddings of the two special tokens <E1> and <E2> as obtained by encoding the utterance with mark-ups using the LLM. Relation embeddings are thus of dimension $2d$ if d is the dimension of the LLM embeddings. For the prototype embedding and comparison metric, we rely on standard variants of few-shot prototype learning as described in [8]. In ProtoNet, aggregation of the K relation embeddings consists in an average pooling, using the cosine distance as metric to compare relation embeddings with prototypes. ProtoNet++ augments this model with unannotated examples, averaging the prototypes with unlabeled relation embeddings weighted by their cosine similarity with the prototype. The MatchingNet extension disregards the prototype construction step and rather compares a query to each utterance in the support set, and the distance to the i-th class of the support set is obtained by averaging the distance to each support utterance embedding e^i_k. Finally the RelationNet model adds to ProtoNet a trainable metric relying on a neural tensor layer [27] to compute the similarity between a query and a prototype.

3.3.2. Choosing the distractors

A key step in defining RaReMUD is the choice of the best distractors relative to a given source relation. This problem differs from the NOTA vector and multiple NOTA vectors approaches in relation classification [23]: although they also model a rejection class, they do not have the notion of source relation and operate in a closed-set scenario. The strategy of RaReMUD consists in selecting adequate distractors among the training relations appearing in $\mathcal{D}_b$ for each source relation to retrieve. The interest of relying on the training data to choose distractors is primarily that the dataset provides a large number of potential distractors to choose from along with a variety of support utterances for each. The simplest approach to consider for the selection of distractors, that will be used as a baseline, consists in making use of all relation types that appear in the training set. The potentially large number of distractors is however likely to be detrimental, both from a detection standpoint and from a computational standpoint. We thus propose and compare selection strategies targeting a limited number R of distractors that are likely to be relevant with respect to a source relation, namely:

1. The *closest distractors* are the R relations whose prototypes are the closest to the source relation's prototype in the embedding space. These distractors are likely to be efficient for fine-grain detection, as they only detect query examples whose embeddings are strongly similar to the source relation's prototype, thus benefiting precision to the expense of recall.
2. In the same logic, the *farthest distractors* are the R relations with the farthest prototypes in the embedding space. Oppositely to the previous strategy, we expect this strategy to focus on eliminating relations rather different from the source one, thus favoring recall to the expense of precision.
3. The *middle distractors* are between the farthest and the closest in the latent space: this strategy might be a middle ground between the fine and coarse-grain detection of the two previous types of distractors, likely to achieve a good balance between recall and precision.
4. *Mixed distractors* are a mix of the 3 previous strategies, possibly benefiting from the combination of their specificities, here again with the idea of achieving a good balance between recall and precision.

We also consider the random choice of R distractors as a naive approach to compare with.

4. Experiments

Extensive experiments are reported to study RaReMUD and the influence of its hyperparameters. We first present the datasets used and provide some details on the implementation of the models and experiments. The experiments that investigate the dependency of RaReMUD's performance on (a) the number of distractors, (b) the distractor selection

strategy and (c) the number of provided examples for a given source relation, are then described. Finally, a post-hoc analysis of our framework compares its current results in terms of F1 score to optimal results.

4.1. Datasets

Experiments are conducted on two of the most recent and popular relation classification datasets available, TACREV [1] and FewRel 1.0 [15]. TACREV offers a variety of relation types, including support utterances bearing no relation. FewRel 1.0 was specially designed for few-shot classification; it offers balanced relation types extracted from Wikipedia with rich textual descriptions. These datasets were preprocessed, removing training utterances that contain entities present in the evaluation set, or in which one of the entities is a possessive adjective (”my”, ”your”, ”her”, ...), as it is unclear whether it refers to the possessor or the possessed object. For both, 75 % of the dataset relation types are kept in $\mathcal{T}_b$ for training and validation (29 relation types for TACREV, 60 for FewRel) and the remaining 25 % make up $\mathcal{T}_n$ (10 for TACREV, 20 for FewRel). Finally, FewRel and TACREV are each divided into three parts: the relation quadruples from $\mathcal{D}_b$ are split into a training set (70 % of these quadruples, i.e., 27,146 for FewRel and 13,012 for TACREV) and a validation set (6,296 for FewRel and 5,436 for TACREV), and the relation quadruples from $\mathcal{D}_n$ make up the evaluation set (13,986 examples for FewRel and 3,325 for TACREV). For each relation type in $\mathcal{T}_n$, three support utterances are selected as examples of the source relation, each being compared to all other utterances of the evaluation set in the retrieval phase. Taking three support utterances per relation type at test time addresses the issue of high intra-class variability induced by the choice of the support example for a given source relation type.

4.2. Implementation details

For all experiments, we use models trained with N=5 and K=5 on a relation classification task in order to learn to build efficient relation prototypes[1]. These small values are justified by the memory-intensive nature of N-way K-shot learning, with each batch comprising at least N*(K+1) utterances (usually N*K*2). Moreover, preliminary experiments pointed that, although the final task is one-shot retrieval, a value of K > 1 for representation learning yields better results. All models were taken from the Hugging Face model repository, using the *roberta-base* pretrained model [19], commonly fine-tuned for various classification and regression tasks, and trained in a parameter-efficient way using low rank adaptation (LoRa) [17]. Early stopping after 3 epochs with no increase in F1 score on the validation set is adopted. The code for the models and experiments of this paper is available in our online repository[2]. For the ProtoNet++ variant that needs extra unlabeled data at training, random sentences are taken from the 2,500 examples of the New York Times dataset available as part of FewRel 2.0 [11] validation split, because of its availability and proximity to our use case’s domain. All evaluations are done by measuring the F1 score of the models on the one-shot relation retrieval task, as an imperfect but effective compromise between precision and recall.

4.3. Impact of the number of distractors

We firstly study the impact of the number of distractors with random selection of the distractors within $\mathcal{T}_b$. The number of distractors can thus range from one to card($\mathcal{T}_b$), the last case corresponding to the baseline as described in Section 3.3.2. For a given source relation and number of distractors, the experiment is repeated 5 times to take into account the variability induced by random choice, using a fixed random seed for the sake of reproducibility. Figure 1 reports the distribution of the F1 score across all source relations as a function of R for the four models on the FewRel dataset—the TACREV dataset displays similar results. On the one hand, a large variability in F1 scores is unsurprisingly observed, explained by the differences between different types of source relations and by intra-class variability due to the choice of the support utterance. We observed that this variability is mainly due to the type of the source relation rather to the choice of the corresponding support utterances which only have moderate influence. For example, with ProtoNet on FewRel and 20 distractors, the highly variable “owned by” and “country” types obtained very low scores (resp. 3 % and 4 %) in contrast to the “crosses” and “league” types (~ 95 %), which are highly specific.

[1] Experiments were carried out using the Grid’5000 testbed (see https://www.grid5000.fr).
[2] https://gitlab.inria.fr/huthomas/raremud

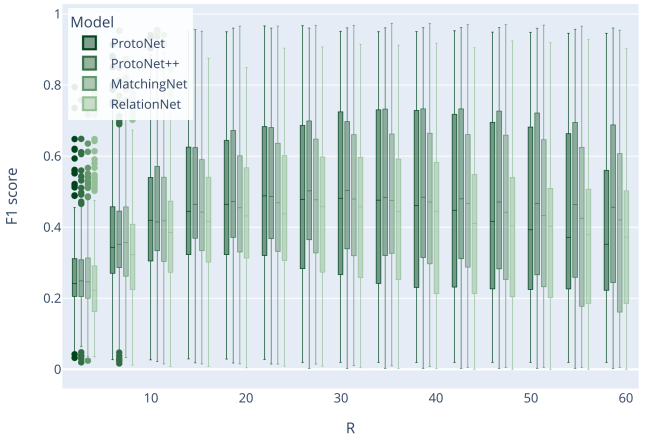

Fig. 1. F1 score of RaReMUD as a function of the amount R of distractors obtained with 4 prototype learning architectures on FewRel.

Globally, medians of F1 scores appear to be relatively low overall though able to reach a decent score ($\sim 50\,\%$) in the best cases. The two previous statements must be looked at against the inherent difficulty of the one-shot relation retrieval task. Nevertheless, it is noticeable that performance starts to improve with the increase of R, then decreases when R becomes too large: keeping all possible distractors, i.e., our baseline, is suboptimal. Although a true general best value is not statistically distinguishable between the high values of R, our analysis indicates a best value around 21 for TACREV and 52 for FewRel, but a finer investigation is needed. According to a Student T-test with the risk of $5\,\%$, these high values are significantly better than those obtained with low values of R and with R=card($\mathcal{T}_b$). This first conclusion that we can draw is that the best general setting regarding the number of distractors is a rather high value for R, however not considering all distractors, typically around 50.

4.4. Comparing distractor selection strategies

Random choice may not be optimal and we now compare the different selection strategies detailed in Section 3.3.2—namely, the (1) closest, (2) farthest, (3) middle and (4) mixed distractors. Table 1 reports the F1 scores for the five distractor selection strategies, averaged over the number of possible distractors from 1 to the maximum number. The strong variability observed in the previous experiments naturally remains in this experiment for the same reasons as mentioned before. A clear advantage is however noticeable when using the middle distractor strategy, with scores significantly better than for other strategies according to a Student T-test with the risk of $5\,\%$, making it a safe overall strategy regardless of the model or dataset. As explained in Section 3.3.2, the good behavior of the middle strategy is likely due to the balance between precision and recall that it offers, strongly rewarded by the F1 score. It is interesting to note that random distractors give good results on average, and even better results than the middle ones for some source relations, but their variance is higher overall, making their use less reliable. For instance, for the relation "country of citizenship" of the FewRel dataset with the ProtoNet model, random distractors obtain an F1 score of 51.54 ± 14.48 whereas middle distractors only obtain 41.15 ± 10.40. This indicates the existence of yet undiscovered distractor choice strategies leading to even better performance, although characterizing these strategies requires further study of distractor choice—see Section 4.6 for more details. We thus recommend the middle distractor strategy as a safe choice for a RareMUD implementation.

4.5. Influence of the amount of annotated source relation examples on performance

Previous experiments were conducted in the extreme context of one-shot relation retrieval, which corresponds to our targeted use-case but is known to be highly challenging. We analyze here the impact of providing more than one

Table 1. F1 scores (in percentages) of RaReMUD with 4 prototype learning architectures on FewRel and TACREV datasets relative to 5 strategies of distractors choice.

	FewRel 1.0				TACREV			
Strategy	ProtoNet	ProtoNet++	MatchingNet	RelationNet	ProtoNet	ProtoNet++	MatchingNet	RelationNet
closest	38.6±25.7	44.4±27.3	39.0±25.4	34.9±20.3	34.2±33.2	27.9±32.0	27.1±29.6	23.6±19.6
farthest	42.6±20.6	41.2±20.2	41.4±21.7	40.2±21.3	35.4±32.6	34.9±32.1	35.0±32.8	30.8±27.8
middle	46.7±22.3	46.2±22.3	44.9±22.8	43.7±21.8	39.3±35.0	37.8±34.1	37.5±35.0	32.8±28.6
mixed	36.9±24.8	43.1±26.1	37.9±24.5	35.0±20.5	33.5±32.2	27.2±30.5	28.7±30.3	25.3±20.7
random	43.7±25.0	46.8±25.4	44.0±24.5	39.2±21.3	37.2±34.3	34.0±34.6	32.9±33.4	28.5±24.1

example. Figure 2 shows the F1 score as a function of the number of utterances describing the source relation, on the FewRel dataset with ProtoNet architecture—other models and the TACREV dataset leading to the same conclusions. A performance gain is clearly visible when two examples are provided instead of one; this performance gain quickly fades when more examples are added. These conclusions, shared by all distractor choice strategies and all datasets, can be applied to our use case, encouraging a journalist to take the time to annotate an additional example in order to visibly improve the retrieval of the source relation in news archives. It also opens perspective for data augmentation with natural language generation, for instance using paraphrasing.

4.6. Analyzing post-hoc optimal distractors

Previous results show that the RaReMUD framework can exhibit convincing average F1 scores given the difficulty of the task, but the distance-based strategies may not provide the best possible choice of distractors. Moreover, if middle distractors work best on average, they can be surpassed by random distractors for some source relations, questioning the RareMUD framework on its upper bound should an optimal choice of distractors be available. As a diagnosis to determine the upper limit of the RaReMUD framework, we thus investigate the ideal configuration of distractors for every source relation based on a post hoc, greedy combinatorial evaluation of distractors: i.e., for each support utterance of each source relation in $\mathcal{T}_n$, we test every possible combination of R = 6 distractors to find the optimal one. The number of distractors is limited to 6 in this case as the greedy search becomes intractable for larger values of R. Results are provided in Figure 3, which plots per source relation the distribution of the scores over the corresponding support utterances for the best possible distractors as determined a posteriori (dark green) and for middle distractors (light green) using the ProtoNet model on FewRel. The number of middle distractors is set to R=55, which has proven a good compromise in the previous experiments. Results demonstrate how powerful the RaReMUD

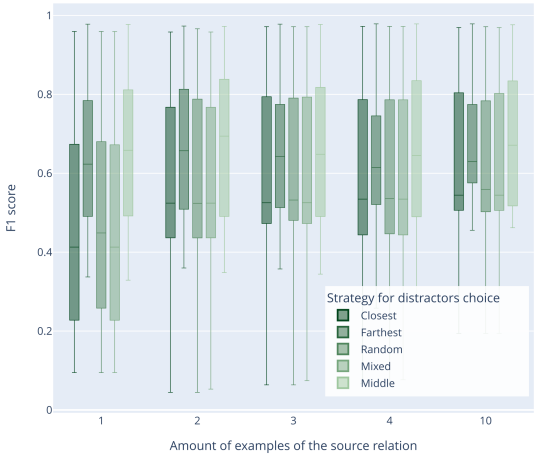

Fig. 2. F1 score of RaReMUD as a function of the amount of examples of the source relation, obtained with 5 distractor choice strategies on FewRel.

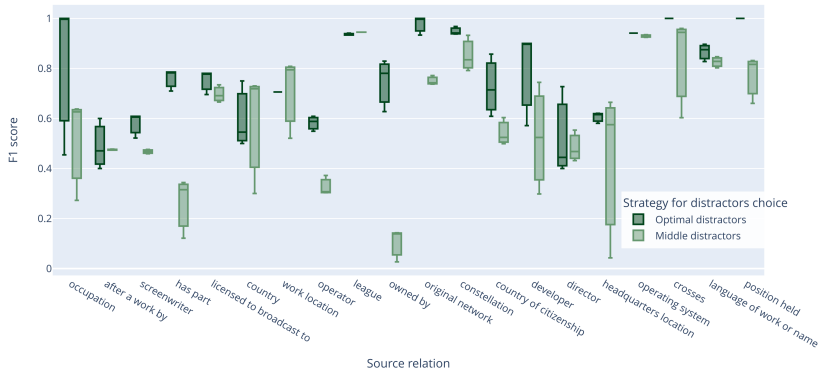

Fig. 3. Comparison of RaReMUD's F1 scores obtained by optimal distractors or middle distractors with the ProtoNet architecture on FewRel.

framework could be if an optimal strategy to choose distractors relevant for a given source relation example were available. Some source relations have moderate scores, even with optimal distractors, proving that either the models have not generalized enough during training to produce faithful prototypes of these novel relations, or that there are no good enough available distractors to suit them. The room for improvement highlighted by our analysis opens the way for future studies, as it remains unclear how to find the optimal distractors for a given source relation when a single example of the relation is available.

5. Conclusion

In this paper, we defined and addressed for the first time the challenging task of one-shot relation mining as a component of an intelligent natural language processing and information retrieval system to find all occurrences of a source relation of interest inside large news archives from a single example. We proposed a framework, RareMUD (Rare Relation Mining Using Distractors), that borrows representation learning to the N-way K-shot rare relation classification scenario and extends it to address single-example relation mining with a rejection class modeled with distractors chosen in a pool of relations. Experimental results reported in the paper firstly demonstrate the effectiveness of the approach to address this highly challenging task with median F-scores around 50 % in the best cases. The ProtoNet and ProtoNet++ variants of the framework also globally provide better results than the MatchingNet and RelationNet ones. We also proposed a safe strategy to select distractors, which consists in mixing a rather large number of distractors that are neither too close nor too far from the source relation example in the relation embedding space learned by the model. While not always optimal, depending on the source relation and the corresponding support, this strategy offers a robust trade-off between recall and precision and is on average better than other variants tested. It should therefore be preferred to implement RareMUD in an actual system. Post-hoc diagnostics experiments highlight that better strategies for a choice of distractors based on the sole supporting example of the source relation remains an issue. Initial experiments with simple indicators—relation frequency, density around the prototype, etc.—have for instance failed to correlate these indicators with the optimal choice. Finally, in all experiments, we observed strong performance variability depending on the source relation and, to a lesser extent, on the corresponding support utterance. We also evidenced that having two examples of the source relation significantly improves retrieval while having more than two is of little interest. These observations call for careful considerations from the user when selecting the source relation example and might impact the design of actual applications where, if possible, two examples should be considered. This can for instance be addressed through iterative search or leveraging paraphrasing models. Actual implementation of an intelligent one or two shot relation retrieval system finally requires more efficient implementation strategies to scale than the N-way K-shot setting adopted in this work at test time. In particular, efficient indexing techniques should be used to avoid the exhaustive comparison of all utterances possibly supporting the source relation in the corpus to the prototypes of the source relation and to the distractors.

References

[1] Alt, C., Gabryszak, A., Hennig, L., 2020. TACRED revisited: A thorough evaluation of the TACRED relation extraction task, in: Annual Meeting of the Association for Computational Linguistics, pp. 1558–1569.

[2] Batista, D.S., Martins, B., Silva, M.J., 2015. Semi-supervised bootstrapping of relationship extractors with distributional semantics, in: Conference on Empirical Methods in Natural Language Processing, pp. 499–504.

[3] Brin, S., 1998. Extracting patterns and relations from the World Wide Web, in: International Workshop on the World Wide Web and Databases, pp. 172–183.

[4] Brody, S., Wu, S., Benton, A., 2021. Towards realistic few-shot relation extraction, in: Conference on Empirical Methods in Natural Language Processing, pp. 5338–5345.

[5] Chen, X., Wu, H., Shi, X., 2023. Consistent prototype learning for few-shot continual relation extraction, in: Annual Meeting of the Association for Computational Linguistics, pp. 7409–7422.

[6] Deepika, S., Geetha, T., 2021. Pattern-based bootstrapping framework for biomedical relation extraction. Engineering Applications of Artificial Intelligence 99.

[7] Detroja, K., Bhensdadia, C., Bhatt, B.S., 2023. A survey on relation extraction. Intelligent Systems with Applications 19.

[8] Dopierre, T., Gravier, C., Logerais, W., 2021. A neural few-shot text classification reality check, in: Conference of the European Chapter of the Association for Computational Linguistics, pp. 935–943.

[9] Gao, T., Fisch, A., Chen, D., 2021. Making pre-trained language models better few-shot learners, in: Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing), pp. 3816–3830.

[10] Gao, T., Han, X., Xie, R., Liu, Z., Lin, F., Lin, L., Sun, M., 2020. Neural snowball for few-shot relation learning, in: AAAI Conference on Artificial Intelligence, pp. 7772–7779.

[11] Gao, T., Han, X., Zhu, H., Liu, Z., Li, P., Sun, M., Zhou, J., 2019. FewRel 2.0: Towards more challenging few-shot relation classification, in: Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing, pp. 6251–6256.

[12] Gong, J., Eldardiry, H., 2023. Prompt-based zero-shot relation extraction with semantic knowledge augmentation. CoRR abs/2112.04539.

[13] Gupta, P., Roth, B., Schütze, H., 2018. Joint bootstrapping machines for high confidence relation extraction, in: North American Chapter of the Association for Computational Linguistics: Human Language Technologies), pp. 26–36.

[14] Han, J., Cheng, B., Lu, W., 2021. Exploring task difficulty for few-shot relation extraction, in: Conference on Empirical Methods in Natural Language Processing, pp. 2605–2616.

[15] Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z., Sun, M., 2018. FewRel: A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation, in: Conference on Empirical Methods in Natural Language Processing, pp. 4803–4809.

[16] Hearst, M.A., 1992. Automatic acquisition of hyponyms from large text corpora, in: International Conference on Computational Linguistics, pp. 539–545.

[17] Hu, E.J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Chen, W., 2021. Lora: Low-rank adaptation of large language models. CoRR abs/2106.09685.

[18] Li, R., Zhong, J., Hu, W., Dai, Q., Wang, C., Wang, W., Li, X., 2024. Adaptive class augmented prototype network for few-shot relation extraction. Neural Networks 169, 134–142.

[19] Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., Stoyanov, V., 2019. Roberta: A robustly optimized BERT pretraining approach. CoRR abs/1907.11692.

[20] Nasar, Z., Jaffry, S.W., Malik, M.K., 2021. Named entity recognition and relation extraction: State-of-the-art. ACM Computing Surveys 54, 1–39.

[21] Niklaus, C., Cetto, M., Freitas, A., Handschuh, S., 2018. A survey on open information extraction, in: International Conference on Computational Linguistics, pp. 3866–3878.

[22] Qu, M., Gao, T., Xhonneux, L.P., Tang, J., 2020. Few-shot relation extraction via Bayesian meta-learning on relation graphs, in: International Conference on Machine Learning, pp. 7867–7876.

[23] Sabo, O., Elazar, Y., Goldberg, Y., Dagan, I., 2021. Revisiting few-shot relation classification: Evaluation data and classification schemes. Transactions of the Association for Computational Linguistics 9, 691–706.

[24] Sainz, O., de Lacalle, O.L., Labaka, G., Barrena, A., Agirre, E., 2021. Label verbalization and entailment for effective zero and few-shot relation extraction, in: Conference on Empirical Methods in Natural Language Processing, pp. 1199–1212.

[25] Snow, R., Jurafsky, D., Ng, A., 2004. Learning syntactic patterns for automatic hypernym discovery, in: Advances in Neural Information Processing Systems, pp. 1297–1304.

[26] Soares, L.B., FitzGerald, N., Ling, J., Kwiatkowski, T., 2019. Matching the blanks: Distributional similarity for relation learning, in: Annual Meeting of the Association for Computational Linguistics, pp. 2895–2905.

[27] Socher, R., Chen, D., Manning, C.D., Ng, A.Y., 2013. Reasoning with neural tensor networks for knowledge base completion, in: Advances in Neural Information Processing Systems, pp. 926–934.

[28] Wadhwa, S., Amir, S., Wallace, B.C., 2023. Revisiting relation extraction in the era of large language modelss, in: Annual Meeting of the Association for Computational Linguistics and International Joint Conference on Natural Language Processing), pp. 15566–15589.

[29] Zhang, Q., Chen, M., Liu, L., 2017. A review on entity relation extraction, in: International Conference on Mechanical, Control and Computer Engineering, pp. 178–183.

[30] Zhong, L., Wu, J., Li, Q., Peng, H., Wu, X., 2023. A comprehensive survey on automatic knowledge graph construction. ACM Computing Surveys 56, 1–62.