

Recherche de relation à partir d'un seul exemple fondée sur un modèle N -way K -shot : une histoire de distracteurs

Hugo Thomas Guillaume Gravier Pascale Sébillot

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

`hugo.thomas@irisa.fr`, `guig@irisa.fr`, `pascale.sebillot@irisa.fr`

RÉSUMÉ

La recherche de relation à partir d'un exemple consiste à trouver dans un corpus toutes les occurrences d'un type de relation liant deux entités dans une phrase, nommé type cible et caractérisé à l'aide d'un seul exemple. Nous empruntons le scénario d'entraînement et évaluation N -way K -shot à la tâche de classification de relations rares qui prédit le type de relation liant deux entités à partir de peu d'exemples d'entraînement, et l'adaptions à la recherche de relation avec un exemple. Lors de l'évaluation, un modèle entraîné pour la classification de relations en N -way K -shot est utilisé, dans lequel $K = 1$ pour le type cible, qui constitue l'une des N classes (du N -way), les $N-1$ classes restantes étant des distracteurs modélisant la classe de rejet. Les résultats sur FewRel et TACREV démontrent l'efficacité de notre approche malgré la difficulté de la tâche. L'étude de l'évolution des performances en fonction du nombre de distracteurs et des stratégies de leur choix met en avant une bonne configuration globale, à savoir un nombre élevé de distracteurs à une distance intermédiaire du type de relation cible dans l'espace latent appris par le modèle. Le diagnostic *a posteriori* de notre méthode révèle l'existence de configurations optimales pour chaque type cible que nos analyses actuelles échouent à caractériser, ouvrant la voie à de futurs travaux.

ABSTRACT

One-shot relation retrieval based on an N -way K -shot model: a matter of distractors

One-shot relation retrieval consists in searching inside a corpus for all occurrences of a relation type linking two entities in an utterance, called the target type and characterized via a single example. In this paper, we perform this task by diverting the training and testing N -way K -shot scenario commonly used for the few-shot relation classification task, i.e., the prediction of the type of relation linking two entities with few training examples. At test time, our approach uses a model trained for N -way K -shot relation classification, in which $K = 1$ for the target type, which constitutes one of the N ways (i.e., N classes), the $N-1$ others being distractors modeling the rejection class. Results on TACREV and FewRel demonstrate the effectiveness of our approach on the challenging task of one-shot relation retrieval. Investigating the influence of the number of distractors used and of their choice exposes an efficient overall configuration, i.e., a high number of distractors at an intermediate distance from the target relation embedding in the latent space learned by the model. Post-hoc analysis of our framework reveals target-type-dependent optimal configurations which our current experiments fail to characterize, paving the way for future work.

MOTS-CLÉS : extraction de relations, apprentissage frugal, apprentissage N -way K -shot.

KEYWORDS: relation extraction, few-shot learning, N -way K -shot learning.

1 Introduction

L'extraction de relations consiste à détecter, dans un texte, les relations liant deux entités (entités nommées ou pronoms leur faisant référence). Conceptuellement, cette tâche se divise en deux sous-tâches : la détection de relations, qui consiste à prédire s'il existe une relation entre deux entités présentes au sein d'une phrase, et la classification de relations, qui suppose l'existence d'une relation et vise à déterminer son type parmi un ensemble prédéfini. En pratique, ces deux sous-tâches sont souvent effectuées conjointement à l'aide d'un modèle qui choisit entre différents types de relations ou une classe de rejet. L'extraction de relations a fait l'objet de nombreux travaux, exploitant notamment des approches neuronales (Zhang *et al.*, 2017; Detroja *et al.*, 2023). Ces avancées restent cependant limitées par la quantité de données annotées nécessaires, conduisant à des performances moindres sur les types de relations rares, c.-à-d. pour lesquels peu d'exemples annotés sont disponibles. Dans cet article, nous nous penchons sur la tâche de recherche de relation avec un exemple. Cette tâche particulière vise à retrouver, dans un grand corpus, toutes les occurrences d'un type de relation cible identifié à l'aide d'un seul exemple. On suppose donc que cet exemple unique suffit à caractériser le type de relation, qui ne possède ni nom ni définition explicites et ne correspond à aucune classe connue d'un modèle. La recherche de relation avec un exemple trouve des cas d'applications concrets par exemple dans un contexte d'analyse de médias, en sociologie ou dans le cadre du journalisme d'enquête, où un utilisateur repère une relation d'intérêt (relation cible) lors de sa recherche et souhaite retrouver d'autres occurrences de ce type de relation dans un corpus.

Dans ce contexte, il n'est pas envisageable de ré-entraîner un nouveau modèle à chaque changement de type de relation cible, pour des questions de temps de calcul et d'absence de données d'entraînement en quantité suffisante. Pour traiter notre problématique, nous nous inspirons des méthodes de classification de relations rares s'appuyant sur des techniques d'apprentissage par transfert (Han *et al.*, 2018) qui permettent d'apprendre à représenter et classifier correctement des types de relations rares grâce à un apprentissage réalisé sur des types de relations fréquents. Ces méthodes sont souvent entraînées et évaluées selon le scénario *N-way K-shot*, dans lequel un *batch* d'entraînement est composé de *N* classes – des types de relations dans notre cas – tirées au hasard parmi des classes connues, chacune d'entre elles étant représentée par un petit nombre *K* de phrases supports issues des données d'entraînement. L'objectif consiste à classifier des énoncés – appelés requêtes – parmi l'une des *N* classes. Ce paradigme d'apprentissage adapté à un faible nombre d'exemples met en avant l'apprentissage de prototypes, c.-à-d. de vecteurs représentant chacun des *N* types de relations et construits à l'aide des *K* exemples disponibles. En pratique, les *N* prototypes d'un *batch* sont utilisés pour classifier chaque requête parmi les *N* types de relations en évaluant la distance – éventuellement apprise lors de l'entraînement – entre ces prototypes et le plongement de la requête appelé vecteur requête. Les modèles réalisant cette procédure sont nommés *modèles d'apprentissage de prototypes*, dont une typologie des architectures est définie par Dopierre *et al.* (2021).

Si le paradigme *N-way K-shot* est particulièrement pertinent pour la classification de relations rares (Han *et al.*, 2018), il ne répond pas directement à notre problématique de recherche de relation à partir d'un seul exemple et doit être fortement adapté pour ce faire. Dans cet article, nous proposons RaReMUD (*Rare Relation Mining Using Distractors*), un modèle dérivé de ce paradigme pour traiter cette tâche. Le modèle est d'abord entraîné pour la classification de relations dans le scénario *N-way K-shot* sur des types de relations fréquents afin d'apprendre à construire des prototypes de relations fiables et, le cas échéant, une métrique de comparaison des plongements (prototypes et vecteurs requêtes). Il est ensuite évalué sur des types de relations rares, toujours dans le scénario *N-way K-shot* : une des *N* classes au sein d'un *batch* correspond au type de relation cible représenté grâce à un seul

exemple support, les (prototypes des) $N-1$ autres classes étant des *distracteurs*¹ modélisant une classe de rejet, c.-à-d. l’absence du type de relation cible. Cette phase d’évaluation correspond à la recherche d’occurrences du type de relation cible dans un corpus, les requêtes d’un *batch* étant à associer soit au type de relation cible, soit à un des distracteurs et par conséquent à rejeter. Nous évaluons plusieurs déclinaisons de notre méthode RaReMUD, incluant différents modèles d’apprentissage de prototypes en N -way K -shot, sur FewRel et TACREV, démontrant son efficacité sur la tâche de recherche de relation à l’aide d’un seul exemple et établissant une référence pour cette tâche. Nous mesurons l’influence des hyperparamètres de RaReMUD sur ses performances afin de déterminer une configuration globale idéale ; nos expériences portent en particulier sur l’impact du nombre de distracteurs utilisés lors de l’évaluation, de la stratégie de leur choix, et du nombre d’exemples pour le type de relation cible. Enfin, le diagnostic *a posteriori* de notre méthode révèle l’existence de configurations optimales pour chaque type cible que nos analyses actuelles échouent à caractériser, mettant en évidence une marge de gain de performances pour RaReMUD.

2 Travaux connexes

À notre connaissance, la recherche de relation à partir d’un seul exemple (ou de peu d’exemples) n’a pas été explorée à ce jour. Cette tâche se différencie de la classification de relations rares par deux aspects principaux : d’une part, c’est une tâche de détection et non de classification ; d’autre part, elle s’effectue en ensemble ouvert, le type de relation cible n’appartenant pas à un ensemble prédéfini. Si la recherche peut être redéfinie en une tâche de classification binaire (détection vs. rejet), la question-clé de la façon de modéliser l’absence de la relation cible demeure.

Malgré ces différences, les travaux les plus proches de notre problématique sont ceux traitant de la classification de relations rares qui visent la prédiction de types de relations parmi un ensemble prédéfini à l’aide de peu d’exemples d’entraînement. Les plus récents s’appuient sur l’apprentissage profond et tirent parti de modèles comme BERT ou GPT, entraînés et évalués dans le scénario N -way K -shot (Han *et al.*, 2021; Qu *et al.*, 2020; Sainz *et al.*, 2021; Gao *et al.*, 2020; Brody *et al.*, 2021; Chen *et al.*, 2023; Li *et al.*, 2024). Dans ce contexte, plusieurs solutions ont été proposées pour obtenir des modèles capables de rejeter des énoncés ne correspondant à aucun des N types de relations présents dans un *batch* : Tan *et al.* (2019) effectuent ce rejet par un apprentissage contrastif ; Gao *et al.* (2019) introduisent la notion de vecteur de rejet en ajoutant un vecteur en plus des N prototypes de relations d’un *batch*. Pendant l’entraînement, ce vecteur est appris directement sans être adossé à des exemples supports, contrairement aux prototypes de relations construits chacun à l’aide de K phrases supports. Cette idée a été étendue par Sabo *et al.* (2021) qui proposent d’apprendre plusieurs vecteurs de rejet pour mieux modéliser cette classe de rejet. RaReMUD emprunte cette idée de vecteurs de rejet multiples pour modéliser l’absence du type de relation cible, mais fonde son modèle de rejet sur des prototypes de relations réelles – appelés distracteurs – construits à l’aide des phrases supports existantes et non sur un apprentissage de paramètres du modèle de classification comme Sabo *et al.* (2021). Ce choix résulte du fait que, dans un contexte d’ensemble ouvert, la classe de rejet dépend du type de relation cible et ne peut être apprise de manière figée.

1. Par souci de simplification, nous employons, dans l’article, indifféremment le terme *distracteur* pour faire référence au type de relation concerné ou à son prototype.

3 Tâche et méthodologie

Dans cette partie, nous introduisons le formalisme de la tâche et rappelons les principes des modèles d'apprentissage de prototypes, avant de détailler notre méthode et nos choix d'implémentation.

3.1 Définition de la tâche

En recherche de relation, les exemples de relations sont des quadruplets de la forme (s, t, r, q) , où la phrase s est support du type de relation $r \in \mathcal{R}$ entre les entités tête et queue $(t, q) \in \mathcal{E}^2$. Les entités considérées sont des entités nommées ou des pronoms leur faisant référence. Par exemple, la phrase $s = \ll \text{La capitale de la Bolivie est Sucre.} \gg$ est support du type de relation $r = \ll \text{capitale de} \gg$ en considérant les entités $t = \ll \text{Sucre} \gg$ et $q = \ll \text{Bolivie} \gg$. On suppose qu'un couple d'entités dans une phrase ne peut être support que d'un type de relation mais qu'il peut y avoir autant de relations dans une même phrase qu'il y a de paires d'entités dans celle-ci. En recherche de relation avec un exemple, un seul quadruplet est disponible pour caractériser le type de relation cible r , la tâche consistant à rechercher tous les quadruplets de ce même type de relation r dans un corpus.

3.2 Apprentissage de prototypes

En apprentissage de prototypes, dans chaque *batch*, un modèle apprend à construire un prototype par type de relation (soit N prototypes) en s'appuyant sur la structure des *batches* du scénario N -way K -shot. Ces derniers sont composés d'un ensemble support noté \mathcal{S} et d'un ensemble requête noté \mathcal{Q} . L'ensemble support $\mathcal{S} = \{s_k^i; i \in [1, N], k \in [1, K]\}$ contient N types distincts de relations, chacun avec K quadruplets supports. L'ensemble requête $\mathcal{Q} = \{s_j; j \in [1, L]\}$ est composé de L phrases et leurs paires d'entités associées, auxquelles on devra attribuer un des N types de relations de l'ensemble support. Les prototypes de relations sont construits grâce aux plongements de leurs exemples supports, appelés vecteurs supports, et chaque requête $s_j \in \mathcal{Q}$ est associée à son plongement, appelé vecteur requête. À l'aide de ces vecteurs, le modèle prend une décision de classification fondée sur la distance entre les vecteurs requêtes et les prototypes. Conceptuellement, les approches d'apprentissage de prototypes requièrent les trois composants suivants, dont les paramètres sont ajustés en utilisant une fonction de coût pour la classification fondée sur l'entropie croisée :

- l'encodage de relations qui représente une phrase et sa paire d'entités par un vecteur de relation : vecteur requête pour toute requête de \mathcal{Q} , et vecteur support (servant à la création d'un prototype) pour chaque support de \mathcal{S} ;
- la construction des prototypes, qui associe K vecteurs issus de phrases supports d'un type de relation de \mathcal{S} à un vecteur unique nommé prototype ;
- la comparaison des vecteurs requêtes et des prototypes fondée sur une mesure de distance.

L'apprentissage N -way K -shot simule une classification de relations rares au niveau de chaque *batch* mais, globalement, le modèle peut être entraîné sur une grande quantité de données annotées, perdant ainsi la notion de rareté des données. En pratique, deux ensembles de types de relations disjoints sont constitués, \mathcal{T}_{fr} et \mathcal{T}_{ra} , et les exemples du jeu de données sont séparés selon ces deux ensembles, formant \mathcal{D}_{fr} et \mathcal{D}_{ra} . Les composants du modèle sont entraînés sur le corpus \mathcal{D}_{fr} , et évalués sur le corpus \mathcal{D}_{ra} afin de mesurer les performances du modèle sur des types de relations rares en pratique au sein de chaque *batch*.

3.3 RaReMUD, une approche N -way K -shot pour la recherche de relation avec un exemple

Notre approche RaReMUD repose sur un modèle d'apprentissage de prototypes entraîné pour la classification de relations en N -way K -shot et l'exploite pour la recherche de relation avec un exemple. Comme décrit dans l'introduction, la phase d'apprentissage considère une tâche classique de classification de relations et permet d'apprendre de manière optimale les composants du système listés ci-dessus. Il est ensuite exploité dans un contexte de recherche de relation à partir d'un exemple, ce dernier étant utilisé comme l'une des classes dans une approche N -way K -shot et complété par $N - 1$ distracteurs modélisant la classe de rejet. Chaque distracteur est construit à l'aide de K exemples supports d'un type de relation fréquent tirés aléatoirement, et est figé durant la recherche ; le choix de fonder les distracteurs sur des types de relations de \mathcal{T}_{fr} est justifié par la nécessité d'avoir plusieurs exemples annotés pour construire des prototypes fiables. Le prototype du type de relation cible, quant à lui, correspond au plongement du seul exemple annoté disponible. Lors de la recherche – correspondant à la phase d'évaluation –, RaReMUD détecte la présence ou l'absence du type de relation cible dans chaque requête, en comparant le vecteur de cette requête au prototype du type de relation cible et aux $N-1$ distracteurs.

Décrire une configuration complète de l'approche RaReMUD nécessite de définir (a) les éléments d'architecture du modèle d'apprentissage de prototypes utilisés et (b) le nombre de distracteurs et la stratégie pour les choisir parmi les types de relations de \mathcal{T}_{fr} . Nous décrivons les quatre architectures étudiées, avant de détailler les choix possibles de distracteurs.

3.3.1 Architectures de modèles d'apprentissage de prototypes

Dans toutes les expériences réalisées, l'encodage d'une phrase et de sa paire d'entités en un vecteur de relation se fait selon Soares *et al.* (2019) : les entités sont délimitées par des balises (<E1> et </E1> pour l'entité tête, <E2> et </E2> pour l'entité queue) ; les *tokens* constituant la phrase sont encodés à l'aide d'un modèle transformeur, et le plongement ou vecteur de relation est obtenu en concaténant les vecteurs des *tokens* <E1> et <E2>. Le reste de l'architecture suit une des variantes de modèles d'apprentissage de prototypes décrites dans Dopierre *et al.* (2021), que nous présentons succinctement. Dans ProtoNet, les N prototypes sont obtenus en effectuant la moyenne des plongements correspondant aux K exemples supports de chaque classe, et la distance cosinus est utilisée pour comparer les vecteurs requêtes aux prototypes. Le modèle ProtoNet++ étend cette idée en améliorant les prototypes grâce à la prise en compte d'exemples non annotés : les vecteurs des exemples non annotés sont ajoutés aux prototypes avec une pondération fondée sur leur similarité cosinus à ces mêmes prototypes. L'approche MatchingNet s'affranchit de la construction des prototypes et compare directement un vecteur requête à chacun des K vecteurs supports d'un type de relation : la distance de la requête à ce type de relation est obtenue en moyennant les distances à ses K vecteurs supports. Enfin, RelationNet ajoute à ProtoNet un apprentissage de métrique s'appuyant sur un modèle neuronal de tenseurs (Socher *et al.*, 2013) en remplacement de la distance cosinus.

3.3.2 Choix des distracteurs

La clé de RaReMUD réside dans le choix, pour un type cible donné, des $N-1$ types de relations de \mathcal{T}_{fr} employés comme distracteurs. Une première approche, qui constituera notre référence, consiste

simplement à prendre tous les types de relations de \mathcal{T}_{fr} comme distracteurs. Cette approche semble cependant peu optimale, notamment en termes de temps de calcul. Nous comparons dans la suite différentes stratégies pour choisir un nombre plus restreint de distracteurs adaptés à un type de relation cible défini par une phrase support et son prototype associé :

- un tirage *aléatoire* des types de relations : cette stratégie constitue une référence faible à titre de comparaison pour les stratégies suivantes ;
- les types de relations dont les prototypes sont *les plus proches* du prototype cible dans l’espace latent construit par le modèle : ces distracteurs permettent *a priori* une détection fine du type de relation cible en rapprochant les prototypes de la classe de rejet de celui de la classe cible ;
- à l’opposé, les distracteurs *les plus éloignés*, c.-à-d. les $N-1$ classes les plus lointaines du type de relation cible dans l’espace latent, représentant un choix moins risqué en assurant de rejeter les exemples éloignés du type de relation cible ;
- des distracteurs *intermédiaires*, situés entre les plus proches et les plus éloignés dans l’espace latent : ce choix constitue potentiellement un juste milieu entre la finesse des distracteurs les plus proches et la sécurité des distracteurs les plus lointains ;
- des distracteurs *mixtes*, mélange à proportions égales des trois choix précédents de distracteurs, en espérant tirer parti du meilleur des trois.

4 Expériences

Les expériences décrites ci-après visent à évaluer RaReMUD et à étudier l’influence de ses hyperparamètres sur ses performances en recherche de relation à partir d’un exemple. Nous décrivons tout d’abord les jeux de données utilisés, puis fournissons des détails d’implémentation des modèles et procédures expérimentales. Nous analysons ensuite, à travers les expériences menées, la dépendance des performances de RaReMUD au choix des distracteurs, plus précisément au nombre de distracteurs employés et aux stratégies pour les choisir parmi \mathcal{T}_{fr} . Nous observons les liens entre le nombre d’exemples annotés pour le type de relation cible et les scores obtenus par RaReMUD, avant de mener une analyse *a posteriori* de notre approche étudiant ses performances dans sa configuration optimale.

4.1 Jeux de données

Les expériences s’appuient sur deux des jeux de données les plus populaires et récents pour la classification de relations : d’une part, TACREV (Alt *et al.*, 2020), une version revisitée de TACRED, qui contient une variété de types de relations, y compris un pour l’absence de relation (ignoré lors de nos expériences) ; d’autre part, FewRel 1.0 (Han *et al.*, 2018), spécifiquement conçu pour la classification de relations rares. Ces jeux de données subissent les prétraitements suivants : les exemples contenant des entités du jeu de données d’évaluation sont supprimés des données d’entraînement, et les exemples (erronés) dans lesquels une entité est seulement constituée d’un adjectif possessif (*his, her, their...*) sont retirés. 75 % des types de relations forment l’ensemble \mathcal{T}_{fr} (soit 60 types pour FewRel et 29 pour TACREV), les 25% restants formant \mathcal{T}_{ra} (20 types pour FewRel et 10 pour TACREV). Au final, FewRel contient 47 428 exemples et TACREV 21 773, séparés en trois parties : les exemples supports de types de relations appartenant à \mathcal{T}_{fr} sont répartis en un jeu d’entraînement (70 % de ces exemples, soit 27 146 pour FewRel et 13 012 pour TACREV) et un jeu de validation (soit 6 296 pour FewRel et 5 436 pour TACREV), et les exemples supports de types appartenant à \mathcal{T}_{ra} constituent le jeu d’évaluation (13 986 exemples pour FewRel et 3 325 pour TACREV). Pour chaque type de \mathcal{T}_{ra} ,

trois exemples supports sont sélectionnés afin de construire indépendamment trois prototypes. Ces prototypes et les distracteurs associés sont chacun leur tour comparés au corpus constitué de tous les autres exemples du jeu d'évaluation.

4.2 Détails des implémentations

Tous les modèles sont entraînés en N -way K -shot pour la classification de relations puis évalués, dans le cadre RaReMUD, pour la recherche de relation à partir d'un (ou des quelques) exemple(s). L'encodage des relations est réalisé par le modèle RoBERTa² (Liu *et al.*, 2019) adapté de manière efficace par LoRA (Hu *et al.*, 2021). L'entraînement est interrompu après trois époques sans amélioration de la mesure F1 sur le jeu de validation. Nous avons expérimentalement fixé pour l'entraînement $N = K = 5$, compromis efficace entre N et K en tenant en compte de la mémoire disponible³. Pour la variante ProtoNet++, le jeu de données *New York Times*, disponible comme jeu de validation de FewRel 2.0, est utilisé comme données non annotées pour augmenter les prototypes lors de l'apprentissage du modèle, du fait de sa disponibilité et de sa relative proximité du domaine des données d'entraînement. En pratique, la phase d'évaluation correspond à un scénario N -way K -shot avec $K = 1$ pour le type cible et $K = 5$ pour les distracteurs, traitant l'ensemble des énoncés requêtes du jeu d'évaluation avec $N = R + 1$.

Pour la stratégie de choix aléatoire des distracteurs, nous effectuons 5 tirages indépendants par expérience avec une graine aléatoire fixe par souci de reproductibilité. La mesure F1, nommée F-mesure par la suite, est la métrique d'évaluation retenue pour juger de la capacité de RaReMUD à retrouver les occurrences du type de relation cible parmi les requêtes, compromis discutable mais efficace entre précision et rappel. Les expériences présentées dans cet article ont été réalisées sur le banc d'essai Grid'5000, soutenu par un groupement d'intérêt scientifique hébergé par l'Inria et comprenant le CNRS, RENATER et plusieurs universités ainsi que d'autres organisations (voir <https://www.grid5000.fr>).

4.3 Influence du nombre de distracteurs

Nous étudions tout d'abord l'influence du nombre de distracteurs, noté R , utilisés lors de la phase d'évaluation. La figure 1 illustre l'évolution de la F-mesure de notre méthode en fonction de R sur FewRel pour les 4 modèles étudiés avec un tirage aléatoire des distracteurs. On constate tout d'abord une très forte variabilité de la F-mesure induite par les différences de scores obtenus selon les types de relations cibles, et par la diversité de prototype du type cible créé selon l'unique exemple annoté servant à le construire. Nous avons observé que cette variabilité est principalement due aux types de relations cibles, la diversité des prototypes ayant une influence plus modérée. Par exemple, avec ProtoNet sur FewRel et 20 distracteurs, les types "owned by" et "country", très variables, obtiennent de très faibles scores (resp. 3 % et 4 %) à l'inverse de "crosses" et "league", très spécifiques (~ 95 %). Par ailleurs, la F-mesure médiane pour chaque valeur de R est, de manière générale, relativement basse. Ces deux points sont à mettre en regard de la difficulté de la tâche de recherche de relation à partir d'un seul exemple. De manière plus fondamentale, il apparaît que la référence ($R = 60$) obtient une meilleure F-mesure qu'un faible nombre de distracteurs mais n'est pas optimale : les valeurs élevées de R inférieures à 60 semblent conduire à de meilleures performances, ce que nous

2. Le modèle `roberta-base` de Huggingface est utilisé.

3. Une implémentation de RaReMUD est disponible à <https://gitlab.inria.fr/huthomas/raremud>.

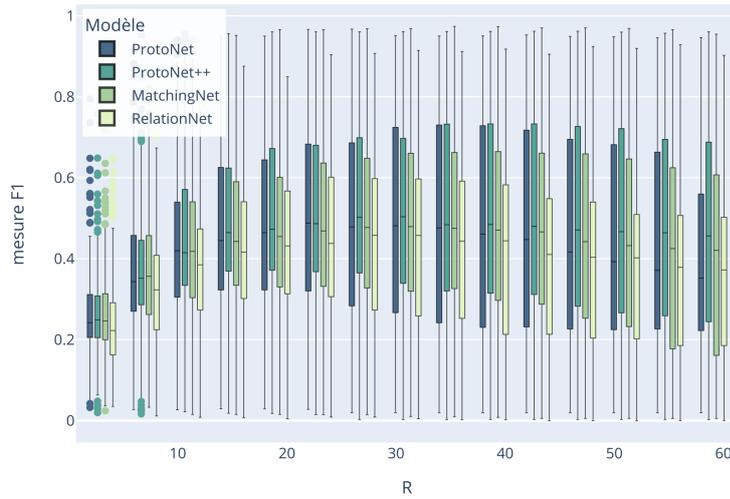


FIGURE 1 – F-mesure sur FewRel en fonction du nombre de distracteurs pour les différents modèles

TABLE 1 – F-mesure (en %) sur FewRel et TACREV pour 5 stratégies de choix des distracteurs.

Stratégie	FewRel 1.0				TACREV			
	ProtoNet	ProtoNet++	MatchingNet	RelationNet	ProtoNet	ProtoNet++	MatchingNet	RelationNet
plus proches	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
plus éloignés	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
intermédiaires	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
mixtes	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
aléatoires	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

avons confirmé par un test statistique de comparaison des moyennes de Student au risque de 5%. Les résultats sur TACREV mènent aux mêmes conclusions. En résumé, RaReMUD obtient donc ses meilleures performances lorsqu'un grand nombre de distracteurs sont choisis parmi tous ceux disponibles, sans que ce nombre soit statistiquement discernable du fait de la grande variabilité des résultats.

4.4 Influence de la stratégie de choix des distracteurs

Nous comparons ensuite les stratégies de choix des distracteurs. Afin de s'affranchir du choix du nombre de distracteurs pour chaque stratégie, nous moyennons les F-mesures obtenues pour une stratégie de choix en faisant varier R . Le tableau 1 rend compte de cette évaluation pour les 4 modèles en fonction de la stratégie utilisée. Malgré la variabilité des résultats (dont les raisons ont été mentionnées en section 4.3), il apparaît que le choix naïf de tirer des distracteurs aléatoires est sous-optimal. La meilleure stratégie moyenne commune à tous les modèles et types de relations cibles semble être celle des distracteurs intermédiaires, constat confirmé par un test de comparaison des moyennes de Student au risque de 5%. Toutefois, certains types de relations cibles obtiennent marginalement de meilleures performances avec des distracteurs aléatoires – le type de relation cible "country of citizenship" de FewRel obtenant p. ex. une F-mesure de $51.54 \pm 14.48\%$ avec les distracteurs aléatoires contre $41.15 \pm 10.40\%$ avec les intermédiaires pour le modèle ProtoNet – ce qui indique que trouver la stratégie optimale pour un type cible donné reste une question ouverte.

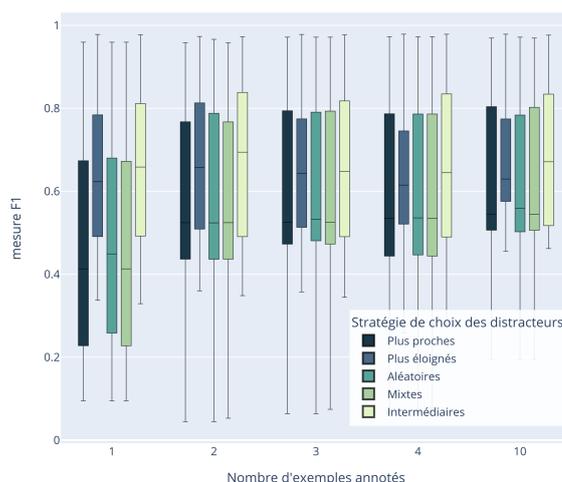


FIGURE 2 – F-mesure de ProtoNet sur FewRel en fonction du nombre d'exemples fournis

4.5 Influence du nombre d'exemples annotés pour la construction de prototype

Les expériences précédentes étant réalisées dans le cas extrême de recherche de relation avec un seul exemple, nous vérifions l'intérêt, par exemple dans un cadre applicatif, d'annoter plus d'un seul exemple pour un type de relation cible. Sur la figure 2, la F-mesure du modèle ProtoNet – l'utilisation des autres modèles menant aux mêmes conclusions – est mise en relation avec le nombre d'exemples fournis pour le type cible sur le jeu de données FewRel. Il apparaît que l'annotation de deux exemples apporte un gain de F-mesure visible par rapport à celle d'un unique exemple. Ce gain s'estompe toutefois rapidement lorsque des exemples supplémentaires sont annotés. Ces conclusions sont communes aux différentes stratégies de choix de distracteurs et au jeu de données TACREV, ce qui les renforce. Dans un cadre applicatif, l'annotation de deux exemples peut donc être recommandée comme un bon compromis entre gain de performance et temps d'annotation.

4.6 Analyse *a posteriori* du choix des distracteurs optimaux

La meilleure configuration globale trouvée jusqu'ici, qui consiste à choisir un nombre élevé de distracteurs à une distance intermédiaire du type de cible, n'est pas nécessairement optimale pour un type de relation cible donné. À des fins de diagnostic de notre approche, nous effectuons l'analyse *a posteriori* suivante : les performances sont évaluées avec toutes les combinaisons possibles de moins de 6 distracteurs – la combinatoire augmentant rapidement avec le nombre de distracteurs – pour chaque type cible afin de déterminer les performances optimales absolues de RaReMUD avec des distracteurs optimaux par type de relation cible. La figure 3 représente la F-mesure sur FewRel de ProtoNet pour plusieurs types de relations cibles avec les distracteurs intermédiaires en quantité R idéale révélée par les expériences de la section 4.3 ($R=55$ pour FewRel) en bleu foncé et avec les 6 distracteurs optimaux déterminés *a posteriori* pour chaque type cible (cette quantité étant choisie empiriquement comme optimale parmi les valeurs disponibles) en vert clair ; L'importante marge de progression entre les deux configurations révèle le potentiel considérable de notre approche sur la tâche pourtant difficile de recherche de relation avec un exemple. Par ailleurs, certains types cibles obtiennent des scores relativement faibles dans les deux configurations, indiquant vraisemblablement

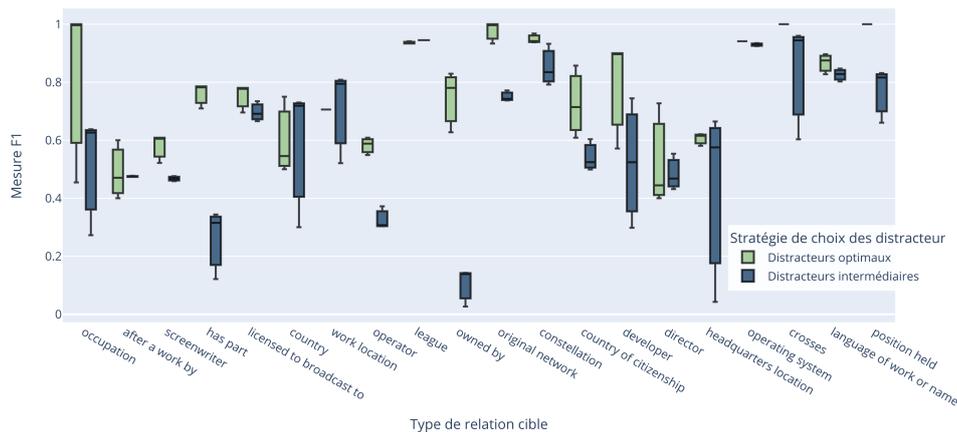


FIGURE 3 – Comparaison des F-mesures avec des distracteurs optimaux et intermédiaires sur les types de relations rares de FewRel pour le modèle ProtoNet

leur difficulté inhérente ou l’absence de distracteurs disponibles adaptés à ces types de relation. Ces deux constats ouvrent la voie à de futurs travaux.

5 Conclusion

Dans cet article, nous avons proposé et étudié RaReMUD (*Rare Relation Mining Using Distractors*), une approche adaptant le scénario de classification de relations rares N -way K -shot à la recherche de relation à partir d’un seul exemple. Nos résultats sur FewRel et TACREV, en fondant RaReMUD sur plusieurs architectures de modèles d’apprentissage de prototypes, démontrent l’efficacité de notre approche sur cette tâche exigeante. Il ressort aussi de nos expériences qu’un choix des distracteurs est nécessaire à l’optimisation de notre méthode, et reste une décision complexe pouvant dépendre du type de relation cible. Nous montrons qu’une stratégie fiable de choix consiste à conserver un nombre de distracteurs élevé, inférieur au nombre maximal, mêlant des types de relations de l’ensemble d’entraînement ni trop proches ni trop éloignés du type de relation cible dans l’espace des prototypes. L’étude de l’influence sur les performances du nombre d’exemples du type de relation cible disponibles souligne que le simple ajout d’un second exemple contribue à augmenter fortement la F-mesure des modèles, encourageant de potentiels travaux exploitant RaReMUD dans un cadre applicatif à annoter au moins deux exemples. Nos expériences révèlent enfin que la stratégie fiable de sélection des distracteurs que nous avons mise en évidence est sous-optimale puisque’une évaluation exhaustive des combinaisons possibles de distracteurs peut conduire à un choix de distracteurs plus performant pour un type de relation cible donné ; la façon de faire ce choix *a priori* en se fondant sur le seul exemple support du type de la relation cible reste toutefois une question qu’il convient d’explorer. Des premières expériences sur des indicateurs simples – fréquence de la relation, densité autour du prototype – n’ont pas permis de corrélérer ces indicateurs au choix optimal des distracteurs. Enfin, à des fins pratiques et de contrôle expérimental, nos expériences ont, jusqu’à présent, été effectuées dans le paradigme de laboratoire N -way K -shot et gagneraient à se rapprocher de conditions d’utilisation réelles. Ceci requiert des approches de fouille plus efficaces que la comparaison exhaustive d’exemples avec N prototypes et des jeux de données dédiés à la tâche de recherche de relations rares, TACREV et FewRel étant conçus pour la classification de relations.

Références

- 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*, p. 1558–1569.
- BRODY S., WU S. & BENTON A. (2021). Towards realistic few-shot relation extraction. In *Conference on Empirical Methods in Natural Language Processing*, p. 5338–5345.
- 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*, p. 7409–7422.
- DETROJA K., BHENSADIA C. & BHATT B. S. (2023). A survey on relation extraction. *Intelligent Systems with Applications*, **19**.
- 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*, p. 935–943.
- GAO T., HAN X., XIE R., LIU Z., LIN F., LIN L. & SUN M. (2020). Neural snowball for few-shot relation learning. In *Conference on Artificial Intelligence*, volume 34, p. 7772–7779.
- 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*, p. 6251–6256.
- HAN J., CHENG B. & LU W. (2021). Exploring task difficulty for few-shot relation extraction. In *Conference on Empirical Methods in Natural Language Processing*, p. 2605–2616.
- 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*, p. 4803–4809.
- 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**.
- 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.
- 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**.
- 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*, p. 7867–7876.
- 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.
- 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*, p. 1199–1212.
- 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*, p. 2895–2905.

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*, volume 26, p. 926–934.

TAN M., YU Y., WANG H., WANG D., POTDAR S., CHANG S. & YU M. (2019). Out-of-domain detection for low-resource text classification tasks. In *Conference on Empirical Methods in Natural Language Processing and International Joint Conference on Natural Language Processing*, p. 3566–3572.

ZHANG Q., CHEN M. & LIU L. (2017). A review on entity relation extraction. In *International Conference on Mechanical, Control and Computer Engineering*, p. 178–183.