《人工神经网络》大作业报告——自主选题

罗宇琦
2019011253
计92
yq-luo19@mails.tsinghua.edu.cn

姚文涛
2019011244
计92
yaowt19@mails.tsinghua.edu.cn

1 简介

选题： 基于跨语言对比学习的低资源语言细粒度实体分类

细粒度实体分类（Fine-grained entity typing, FGET）的目标是将命名实体分类成细粒度的实体类型，该任务对于许多和实体相关的NLP任务都很有意义。FGET面临的一个主要问题是低资源问题，复杂实体类型层次结构增加了手动标注数据的难度，特别是除英语之外的语言，有标注数据集非常稀少。为此我们提出了一个跨语言对比学习框架Cross-C，来解决低资源语言的FGET问题。我们使用预训练模型BERT[1]的多语言版本m-BERT作为backbone，来帮助实体分类知识从高资源语言（例如英文）迁移到低资源语言（例如中文）。此外，我们利用基于实体对的远程监督假设[2]以及机器翻译来获得跨语言的远程监督数据，并在远程监督数据上进行跨语言对比学习。实验表明，通过应用我们的方法，即便在没有任何低资源语言的有标注数据的情况下，也可以轻松训练得到适用于低资源语言的FGET模型。

2 相关工作

作为信息抽取领域最重要的任务之一，对于FGET的研究由来已久。由于细粒度类型为语言理解带来了更精细的语义信息，因此这些类型被广泛用于提升实体相关的NLP任务的表现，如共指解析、实体链接、关系抽取和事件抽取等。

由于实体类型具有复杂的层次结构，手动注释FGET数据并不容易，因此低资源问题是FGET的关键挑战之一。为了缓解这一问题，远程监督方法已被广泛用于FGET。一种典型的远程监督方法是使用知识库来自动注释文本中提到的实体。Ling[3]和Gillick[4]收集维基百科页面中与知识库中实体相对应的锚点，并用知识库中的实体类型标记这些锚点。这种方法被后续的一系列工作采用，以获得类型伪标签。我们的工作也借鉴了这一思路。其他方法还有使用句子中的各种名词短语作为类型的伪标签等。

FGET数据集的构建也在不断推进。早期的实体分类数据集CoNLL[5]和Ontonotes[6]只覆盖了几种粗粒度类型。此后，Choi[7]通过引入包含数千种类型的超细集合，进一步扩展了FGET，虽然其中许多超细类型只有远程监督的样本。Few-NERD[8]引入了大约一百种细粒度类型，并为每种类型注释了大量样本。然而，上述工作主要局限于英文领域。也有一些工作建立了其他语言的FGET数据集，如中文、日文、荷兰语和西班牙语，但这些非英文数据集的规模和质量远无法与英语数据集相比，可以认为，大多数非英文语言的有标签实体分类数据仍处于低资源的窘境中。

虽然跨语言学习在实体链接和命名实体识别方面已经得到了广泛的探索，但是跨语言实体分类方面则没有充分的工作。Selvaraj [9]基于预训练模型M-BERT提出了可以迁移到低资源语言的跨语言细粒度实体分类模型，但其没有使用对比学习方法。对于对比学习，一些初步探索将其用于实体关系抽取 [10]，并取得了可观的结果。Peng[11]进一步使用对比学习来分析实体信息对关系抽取的影响。对比学习在FGET任务上的探索仍处于初级阶段，缺乏有力的工作探索在低资源语言FGET任务上的作用。

3 方法

3.1 基础模型

本文提出的方案是一种对模型进行预训练的通用方案，因此可以自由选择基础模型，在我们的实验中主要使用了BERT multilingual base model（以下简称m-BERT）进行预训练以及微调来验证效果。m-BERT是BERT模型的一个多语言版本，使用了104种语言进行预训练，使得m-BERT具有较好的跨语言性能。

3.2 远程监督

考虑远程监督假设，若句子S_1和句子S_2中都出现了A和B两个实体提及，那么我们认为句子S_1、S_2中的A的类型是相同的，同时S_1、S_2中的B的类型也是相同的。另外，我们还借助机器翻译进行了跨语言的远程监督，例如有英文句子S_E，其中出现了实体A，我们使用机器翻译将该句子翻译成中文S_C，并设法定位实体A翻译之后的位置，那么可以认为S_E，S_C中的A的类型是相同的。

3.3 实体遮蔽

由于使用上述远程监督得到的句子对中实体的名称将完全一样，为了提高模型利用上下文信息的能力，我们应用了实体遮蔽方法。例如句子“**北京**是中国的首都”，在训练的时候，有0.5的概率将“北京”用一个表示遮蔽的符号代替。这样可以使得模型在训练的时候学习如何利用上下文信息，而不会仅仅关注于实体名称。

3.4 正例及负例的获得

假设现在有一个句子S，其中有n个标记出来的实体$e_1, e_2, \cdots, e_n$，那么我们会根据这个句子构造出n个训练用例，其中第i个用例包含句子S以及被标记的第i个实体e_i，并且具有$n-1$个标签：$e_i \& e_j$, $\forall i \neq j$，其中"&”用于分割两个实体名称。

对于跨语言交叉对比学习部分，每个用例都会具有一个标签：translate#i，其中i为该用例的编号，从而可以方便地找到一个用例翻译之后对应的用例。

训练时，首先选取k个训练用例，之后对于每个训练用例，在所有与其具有至少一个相同标签的用例中随机选取一个作为其正例，这样一共得到$2k$个用例，其中每个用例拥有1个正例，并且将剩余的$2(k-1)$个用例作为负例，之后输入模型进行对比学习。

3.5 模型的结构及输入输出

模型的结构主要包括两部分：作为编码器的m-BERT以及作为分类器的线性层L。

以“北京是中国的首都”中对“北京”的实体分类为例，输入m-BERT的文本为“<ent>北京<ent>是中国的首都”，其中<ent>为我们自定义的token，用于在输入文本中标记出实体的位置。m-BERT会为文本中的每个token都生成一个词向量，我们取第一个<ent>的词向量作为“北京”的实体分类的向量表示x。之后，x将通过分类器L映射到一个维数为类别标签个数的向量，从而进行实体分类。

3.6 预训练过程

预训练的目的是让相同类型的实体在经过m-BERT编码之后的向量表示尽可能接近，而非相同类型的尽可能远离。设$f(\cdot,\cdot)$为衡量两个向量表示的相似度的函数，那么若实体i和实体j类型相同，则$f(x_i, x_j)$应当尽量大，反之则应当尽量小，为此我们应用了对比学习的方法。

在预训练过程中，首先使用3.4中的采样方法选取得到m个用例，然后将每个用例都输入模型，根据3.5中的描述可以得到每个用例的实体分类的向量表示$x_1, x_2, \cdots, x_m$，之后应用损失函数计算得到loss，就可以进行预训练了。

对比学习的损失函数表示如下：

$$L_{contrastive} = \sum_{i=\mathbf{1}}^{m} -\mathbf{log}\left(\frac{\sum_{j=\mathbf{1},j\neq i}^{m} e^{f(x_i,x_j)} \cdot g(i,j)}{\sum_{j=\mathbf{1},j\neq i}^{m} e^{f(x_i,x_j)}}\right)$$

其中x_i, x_j表示模型的输出，$g(i,j)$表示第i个用例和第j个用例之间是否为正例，若是则$g(i,j) = 1$，否则$g(i,j) = 0$。那么在优化该损失函数的过程中就能够提高正例之间的相似度，同时抑制负例之间的相似度。

由于最后的分类器是一个线性层，因此选取两个向量表示的余弦距离作为它们的相似度是较为合理的选择，又因cos函数的值域为$[-1,1]$较为窄，还需要乘一个系数来放大以便于模型区分，最终相似度函数表示如下：

$$f(x_i, x_j) = \frac{x_i^T x_j}{|x_i|\,|x_j| \cdot \tau}$$

其中τ表示"温度"，温度越低则模型越容易区分不同的实体类型。在我们的实验中取$\tau = 0.5$。

3.7 Fine-tuning过程

预训练后得到的预训练模型可以被用于具体实体分类任务，只需要修改线性层L的输出维度大小为类型数量，之后进行Fine-tuning即可。

令$z_i = L(x_i)$表示模型的最终输出，y_i为用例i的类型的one-hot向量（如果有多个类型则为多个one-hot向量之和），Fine-tuning的损失函数表示如下：

$$L_{fine-tuning} = -\sum_{j=1}^{l} y_i(j) \cdot \log\ \sigma(z_i(j)) + (1 - y_i(j)) \cdot \log\ \sigma(1 - z_i(j))$$

其中l为类型的数量，σ为sigmoid函数。

4 实验

在本节中，我们评估了我们的框架在两个典型的关于实体的数据集上的有效性:Open-Entity和Few-NERD。对于每个数据集，我们分别在低资源（few-shot/zero-shot）和完整数据集（full-set）设置下进行了实验。除了定量实验，为了进一步展示我们的方法的运作机制，我们进行了消融实验并对特征空间进行了可视化，以便进行定性分析。

4.1 数据集设置

Open-Entity [7]和**Few-NERD** [8] 都是很受欢迎的FGET数据集。Open-Entity包括9种通用类型和121种细粒度类型，其中的每个实体样本可能对应于多个实体类型。Few-NERD包括8种通用类型和66种细粒度类型。这两个数据集都有清晰的类型层次结构，非常适合评估模型在实体分类任务上的性能。在我们的实验中，我们要求模型预测句子中包含的每个实体的通用类型和细粒度类型。

4.2 实验设置

在本课题中，我们选择英语作为高资源语言，汉语作为低资源语言。我们希望仅使用人工标注的英文数据和大规模未标注多语言数据进行学习，用以获得一个有效的中文FGET模型。这是非常困难的，因为在这个过程中没有使用任何人工标记的中文数据。Open-Entity和Few-NERD的所有测例也被机器翻译为中文，用于模型评估。

为了获得远程监督数据，我们使用启发式规则对英文和中文的维基百科页面 [1]进行自动标注。随后，我们使用机器翻译 [12]将英语远程监督样本翻译为相应的中文样本，用于跨语言对比学习。

[1] https://dumps.wikimedia.org/

Dataset	Model	2-Shot			4-Shot			8-Shot			16-Shot		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Open-Entity	F-T	71.4	20.8	32.2	69.4	26.4	38.3	71.3	30.3	42.5	71.3	45.8	55.8
	MONO-C	48.9	38.6	$43.1^{\uparrow 10.9}$	51.2	45.7	$48.3^{\uparrow 10.0}$	56.7	49.8	$53.1^{\uparrow 10.6}$	63.5	58.6	$60.9^{\uparrow 5.1}$
	CROSS-C	56.4	42.0	$\mathbf{48.1}^{\uparrow 15.9}$	58.3	43.8	$\mathbf{50.1}^{\uparrow 11.8}$	60.7	51.1	$\mathbf{55.5}^{\uparrow 13.0}$	70.2	59.9	$\mathbf{64.6}^{\uparrow 8.8}$
Few-NERD	F-T	72.7	25.1	37.3	73.2	35.7	48.0	71.8	44.1	54.7	69.2	51.7	59.2
	MONO-C	54.2	41.7	$47.2^{\uparrow 9.9}$	64.3	51.2	$57.0^{\uparrow 9.0}$	65.9	56.4	$60.8^{\uparrow 6.1}$	67.8	60.4	$63.9^{\uparrow 4.7}$
	CROSS-C	56.4	45.7	$\mathbf{50.5}^{\uparrow 13.2}$	66.3	56.3	$\mathbf{60.9}^{\uparrow 12.9}$	70.3	62.4	$\mathbf{66.1}^{\uparrow 11.4}$	69.9	66.0	$\mathbf{67.9}^{\uparrow 8.7}$

Table 1: 中文测试集上的模型性能(%)。所有模型都在few-shot下训练。K-Shot意为每个实体类型只有K个样本用于训练。↑表示与F-T相比，模型性能的提升量。

Dataset	Model	2-Shot			4-Shot			8-Shot			16-Shot		
		P	R	F_1	P	R	F_1	P	R	F_1	P	R	F_1
Open-Entity	F-T	69.2	35.7	47.1	67.5	43.1	52.6	68.7	49.6	57.6	65.9	55.5	60.3
	MONO-C	59.1	44.3	$50.6^{\uparrow 3.5}$	57.8	50.2	$53.7^{\uparrow 1.1}$	59.6	56.3	$57.9^{\uparrow 0.3}$	61.1	60.9	$\mathbf{61.0}^{\uparrow 0.7}$
	CROSS-C	56.8	45.9	$\mathbf{50.8}^{\uparrow 3.7}$	61.0	51.7	$\mathbf{55.9}^{\uparrow 3.3}$	61.6	58.2	$\mathbf{59.8}^{\uparrow 2.2}$	59.5	62.1	$60.8^{\uparrow 0.5}$
Few-NERD	F-T	78.4	38.3	51.4	79.2	49.6	61.0	80.0	61.4	69.5	78.4	67.9	72.8
	MONO-C	56.3	48.3	$52.0^{\uparrow 0.6}$	63.0	61.4	$62.2^{\uparrow 1.2}$	76.9	70.8	$\mathbf{73.7}^{\uparrow 4.2}$	78.7	70.2	$74.2^{\uparrow 1.4}$
	CROSS-C	66.6	52.0	$\mathbf{58.4}^{\uparrow 7.0}$	73.8	63.5	$\mathbf{68.3}^{\uparrow 7.3}$	75.9	69.3	$72.4^{\uparrow 2.9}$	78.8	73.1	$\mathbf{75.9}^{\uparrow 3.1}$

Table 2: 英文测试集上的模型性能(%)。所有模型都在few-shot下训练。K-Shot意为每个实体类型只有K个样本用于训练。↑表示与F-T相比，模型性能的提升量。

我们共使用了三种典型的实验设置:

少次学习（Few-shot）。此设置要求模型仅在少量有监督数据上训练，并推断实体类型。我们为每个实体类型随机抽取2、4、8、16个例子进行训练。

零次学习（Zero-shot）。此设置要求模型在未进行任何有监督训练的情况下推断实体类型，即没有人工标记样本用于训练。

全样本学习（Full-set）。在此设置中，数据集内所有的有监督样本都用于训练。

我们使用F1值来评估模型的性能。

4.3 Baseline设置

我们使用M-BERT [1]作为backbone结构[2]来实现所有baseline和我们的模型**CROSS-C**。各个模型最后都使用英文人工标注数据对预训练模型进行Fine-tuning。**F-T**不使用对比学习进行预训练，而直接对M-BERT进行Fine-tuning。**MONO-C**表示仅使用英文数据的单语言对比学习进行预训练，再进行Fine-tuning。我们的模型 **CROSS-C**则同时使用了中文和英文及其翻译进行跨语言交叉对比学习。上述所有模型均采用AdamW优化，学习率为{5e-6，1e-5，3e-5，5e-5}。用于预训练和Fine-tuning的batch size为{8，16，32，64，128，256}。对于跨语言对比学习，我们只对大规模远程监督数据遍历一次；在有标签数据上Fine-tuning时，epoch则分别为{1，3，5，7，10}。计算余弦相似度使用的temperature为0.5。

4.4 低资源条件下的整体性能

Few-shot设置下的中文实体分型结果见表1。该表显示:

(1)使用多语言PLM作为backbone可以为低资源语言训练出有效的FGET模型。实验中的所有方法，包括baseline模型和我们的CROSS-C模型，并不需要使用任何中文的人工标记训练数据，就可以在中文测试集中获得有效的实体分类结果。

(2)使用远程监督数据进行对比学习可以显著提高backbone PLM的实体分类能力。与在高资源语言中直接用人工标注数据对多语言PLM进行微调相比，对多语言的远程监督数据进行对比学习可以更好地连接高资源语言和低资源语言，这对于在低资源语言中获得有效的模型非常有利。

[2] https://github.com/google-research/bert

Model	Open-Entity (Chinese)		
	P	R	F_1
M-BERT	8.8	4.0	5.5
CROSS-C	**24.3**$^{\uparrow 15.5}$	**11.4**$^{\uparrow 15.3}$	**15.5**$^{\uparrow 10.0}$

Model	Open-Entity (Chinese)		
	P	R	F_1
M-BERT	6.2	3.5	4.5
CROSS-C	**25.5**$^{\uparrow 19.3}$	**13.5**$^{\uparrow 10.0}$	**17.7**$^{\uparrow 13.2}$

Table 3: 中文测试集上zero-shot设置下的模型性能(%)。所有模型都不使用任何有标签样本对模型进行调整。↑表示与M-BERT相比，模型性能的提升量。

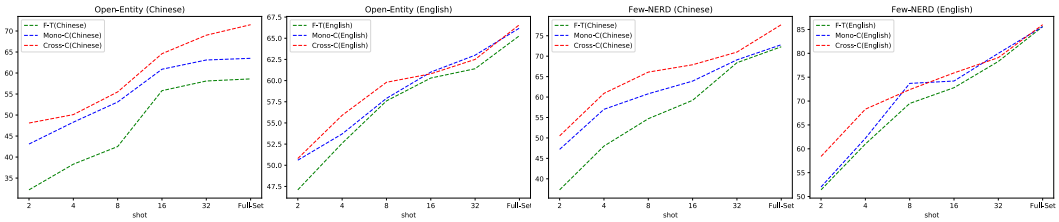

Figure 1: 模型性能(%)随有监督样本数量的变化曲线。我们分别汇报了模型在中文及英文测试集上的F_1值(%)。

(3)与单语对比学习相比，我们的跨语言对比学习可以更好地促进实体类型知识从高资源语言向低资源语言的转移。我们的CROSS-C在zero-shot、one-shot和full-set设置中都能获得最佳性能，且性能提升幅度随着有监督样本的减少而逐渐增加。该结果表明，即使没有任何的低资源语言的高质量有标签数据，使用我们的方法可以有效地提高低资源语言实体分类任务上的模型性能。

我们还在表2 中汇报了模型在原始英文测试集上的实体分类性能。从表中我们可以看到:

(1)在低资源设置下的实验中，中文完全没有或只有少量人工标注的数据，然而，每个实体类型都有一些高质量的英文数据。因此，对比学习在英语测试集上的表现提升效果不如在中文测试集上明显。然而，与直接微调PLM相比，对比学习方法仍然带来了一些性能提升，证明了远程监督方法在数据增强上的能力。

(2)即使在英文实体分类任务中，由于多语言数据可以使多种语言相互学习，跨语言对比学习相对于单语言对比学习也有额外的性能提升。这证明了我们的跨语言对比学习框架的有效性。

表3 显示了中文测试集上zero-shot实体分类的结果。在这张表中，我们可以看到:当我们没有训练类型分类器时，跨语言对比学习在预训练阶段给backbone PLM带来了很强的类型识别能力。这也是我们的跨语言对比学习方法能够处理低资源语言的本质原因。

4.5 全样本条件下的整体性能

图1 展示了模型性能随有监督样本数量的变化曲线。值得注意的是，增加的仅仅是高资源语言（英文）的有监督样本数，而低资源语言（中文）仍然没有人工标注数据。图中的结果显示:

(1)对于高资源语言，随着有监督样本的增加，对比学习带来的性能改善逐渐减少，这符合我们的直觉。但我们也应注意到，即使在full-set设置下，对比学习方法也能获得与对PLM进行Fine-tuning相当、甚至微略更好的结果。这意味着采用对比学习可以很好地降低数据噪声的不利影响，同时通过充分利用远程监督数据来提高模型性能。

(2)不论是在低资源设置还是full-set设置下，我们的跨语言对比学习模型在中文测试集上的结果总是显著高于其他baseline模型。这表明我们的框架可以利用高资源语言的有标签数据和大规模的多语言无标签数据来完成低资源语言的FGET任务。

Model	Open-Entity (Chinese)			
	2-Shot	4-Shot	8-Shot	16-Shot
CROSS-C	**48.1**	**50.1**	**55.5**	**64.6**
−cc	$45.2^{\downarrow 2.9}$	$46.9^{\downarrow 3.2}$	$48.1^{\downarrow 7.4}$	$55.7^{\downarrow 8.9}$
−cc−zc	$43.1^{\downarrow 5.0}$	$48.3^{\downarrow 1.8}$	$53.1^{\downarrow 2.4}$	$60.9^{\downarrow 3.7}$
−cc−zc−ec	$32.2^{\downarrow 15.9}$	$38.3^{\downarrow 11.8}$	$42.5^{\downarrow 13.0}$	$55.8^{\downarrow 8.8}$
Model	Few-NERD (Chinese)			
	2-Shot	4-Shot	8-Shot	16-Shot
CROSS-C	**50.5**	**60.9**	**66.1**	**67.9**
−cc	$48.1^{\downarrow 2.4}$	$59.2^{\downarrow 1.7}$	$63.2^{\downarrow 2.9}$	$64.9^{\downarrow 3.0}$
−cc−zc	$47.2^{\downarrow 3.3}$	$57.0^{\downarrow 3.0}$	$60.8^{\downarrow 5.3}$	$63.9^{\downarrow 4.0}$
−cc−zc−ec	$37.3^{\downarrow 13.2}$	$48.0^{\downarrow 12.9}$	$54.7^{\downarrow 11.4}$	$59.2^{\downarrow 8.7}$

Table 4: CROSS-C的消融实验结果。我们汇报了在few-shot设置下得到的F_1值(%)。↓表示在弃用一些对比学习目标后，模型性能相比CROSS-C的下降量。

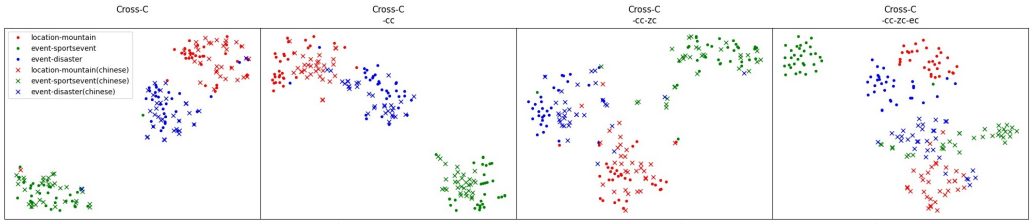

Figure 2: CROSS-C消融实验中的分类效果可视化。我们选取了Few-NERD数据集中一些典型的实体类型及其对应的样本用于可视化，其中·为英文实体样本，×为中文实体样本。

4.6 消融实验与结果可视化

为了更直观地展示我们的跨语言对比学习模型Cross-C的工作机制，我们进行了全面的消融实验。实验结果如表4所示，其中“−cc”表示我们移除了用于预训练backbone PLM的跨语言对比学习目标，“−zc”表示我们移除了在中文远程监督数据上的单语言对比学习目标，“−ec” 则意味着我们移除了对英文远程监督数据的单语言对比学习目标。从表4中我们可以发现：单语言对比目标和跨语言目标在增强backbone PLM中都起到了重要作用，两者结合时则可以带来更大的提升。正因如此，我们的跨语言对比学习模型在对backbone进行预训练时，既要包括单语言对比目标，也要包括跨语言对比目标。

我们还在图2中给出了跨语言对比学习模型CROSS-C在消融实验过程中的分类效果可视化。可视化结果显示，如果不进行任何对比学习，很难将高资源语言和低资源语言中的实习类型知识联系起来。随着我们逐步加入单语言和跨语言的对比学习目标，实体类型之间的区分效果变得更好（尤其对于低资源语言），不同语言之间的语义融合也变得更好。

5 总结

我们提出了一个有效的跨语言对比学习框架，可以通过预训练，在同样的特征空间中同时学习多语言的语义表示。我们使用了基于实体对的远程监督原则以及机器翻译来自动获得高质量的跨语言数据，然后应用对比学习来增强模型的能力，将高资源语言中的实体分类知识迁移到低资源语言中，并通过实验验证了该方法在无需低资源语言的人工标注数据的情况下，能够为低资源语言的FGET任务带来显著的性能提升。

在未来的工作中，我们希望探索如何更好地利用无监督数据来解决FGET的低资源问题。此外，跨语言的FGET数据集标注也是一个有价值的探索方向，这可以推动FGET在英语之外的低资源语言（例如中文）中的发展。

代码下载：https://git.tsinghua.edu.cn/yq-luo19/entity_typing/-/tree/master

参考文献

References

[1] Devlin, J., M.-W. Chang, K. Lee, et al. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of NAACL-HLT*, pages 4171–4186. 2019.

[2] Mintz, M., S. Bills, R. Snow, et al. Distant supervision for relation extraction without labeled data. In *ACL 2009, Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP, 2-7 August 2009, Singapore*. 2009.

[3] Ling, X., D. S. Weld. Fine-grained entity recognition. In *Proceedings of AAAI*, pages 94–100. 2012.

[4] Gillick, D., N. Lazic, K. Ganchev, et al. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*, 2014.

[5] Sang, E. T. K., F. De Meulder. Introduction to the conll-2003 shared task: Language-independent named entity recognition. In *Proceedings of NAACL-HLT*, pages 142–147. 2003.

[6] Hovy, E., M. Marcus, M. Palmer, et al. Ontonotes: the 90% solution. In *Proceedings of NAACL-HLT*, pages 57–60. 2006.

[7] Choi, E., O. Levy, Y. Choi, et al. Ultra-fine entity typing. In *Proceedings of ACL*, pages 87–96. 2018.

[8] Ding, N., G. Xu, Y. Chen, et al. Few-nerd: A few-shot named entity recognition dataset. In *Proceedings of ACL*, pages 3198–3213. 2021.

[9] Selvaraj, N., Y. Onoe, G. Durrett. Cross-lingual fine-grained entity typing. *arXiv preprint arXiv:2110.07837*, 2021.

[10] Soares, L. B., N. FitzGerald, J. Ling, et al. Matching the blanks: Distributional similarity for relation learning. In *Proceedings of ACL*, pages 2895–2905. 2019.

[11] Peng, H., T. Gao, X. Han, et al. Learning from context or names? an empirical study on neural relation extraction. In *Proceedings of EMNLP*, pages 3661–3672. 2020.

[12] Klein, G., Y. Kim, Y. Deng, et al. Opennmt: Open-source toolkit for neural machine translation. In *Proceedings of ACL, System Demonstrations*, pages 67–72. 2017.