# PRACTICAL NAMED ENTITY RECOGNITION: THE ROLE OF ENTITY AND ITS CONTEXT

Oshin Agarwal

A DISSERTATION

in

Computer and Information Science

Presented to the Faculties of the University of Pennsylvania

in

Partial Fulfillment of the Requirements for the

Degree of Doctor of Philosophy

2023

Supervisor of Dissertation

Ani Nenkova, Principal Scientist at Adobe Research

Graduate Group Chairperson

Mayur Naik, Professor of Computer and Information Science at University of Pennsylvania

Dissertation Committee

Dan Roth, Eduardo D. Glandt Distinguished Professor of Computer and Information Science at University of Pennsylvania

Mark Y. Liberman, Christopher H. Browne Distinguished Professor of Linguistics, Professor of Computer and Information Science at University of Pennsylvania

Mark J. Steedman, Professor of Cognitive Science in the School of Informatics at University of Edinburgh, Adjunct Professor of Computer and Information Science at University of Pennsylvania

Daniel M. Bikel, Senior Staff Research Scientist at Meta AI

PRACTICAL NAMED ENTITY RECOGNITION: THE ROLE OF ENTITY AND ITS CONTEXT

*Dedicated to Raj.*

# ACKNOWLEDGEMENTS

This thesis is a joint effort that would not have been possible without the support of numerous people.

First and foremost, I thank my thesis advisor Ani Nenkova. I could not have asked for a better advisor. It is her invaluable advice on research, career and even life that has allowed me to grow to the place I am today. Her advice and feedback helped me improve and develop my skills over time. When I was tangled up in too many numbers and details, her advice to not only take a step back to look at the bigger picture but also how to structure my thoughts to do so has helped me a lot. At the same time, her words of appreciation and encouragement made me feel more confident in myself. I will never forget her help with my internship and job search and the kind words about me when making introductions. She cares a lot about her students and values mental health and time with family. Recognizing the importance of these is one thing but she goes above and beyond to make sure that you are doing well. I truly appreciate how she encouraged me to take breaks, suggested activities and told me to not worry about work when I visited my parents. I am grateful and honoured to have had her as my advisor. I am going to miss our weekly meetings immensely.

I also thank my thesis committee, Dan Roth, Mark Liberman, Mark Steedman and Dan Bikel, for their insightful comments and questions. They have helped refine the thesis and shape its direction. Their suggestions to include figures and definitions have greatly improved the presentation, readability and clarity of the thesis.

I had several mentors through the doctoral program who supported me. I especially thank Byron Wallace and Yinfei Yang. They were my first two collaborators after I joined the program and I wrote my first few papers with them. They have continued to provide me with support and advice. Yinfei also helped me in the search for internships and in fact hosted me for one of them.

I would like to thank the mentors and collaborators I met during my internships. I thank Dan Bikel who gave me a chance and hosted me for my first internship in the program. Every other internship and several opportunities that followed would not have been possible without it. I learned so much from him and appreciate him for serving on my thesis committee as well. I also thank my other internship mentors and collaborators, Heming Ge, Rami Al-Rfou, Siamak Shakeri, Gustavo Hernandez Abrego, Mandy Guo, Jianmo Ni, YunHsuan Sung, Yinfei Yang, Shyam Upadhyay and Tong Sun. They taught me so much about the field beyond just my thesis and made the internships very memorable despite being virtual.

I thank the CIS department chair Zack Ives, graduate group chairs Rajeev Alur and Mayur Naik, graduate coordinator Britton Carnevali, Tori Frew from RAS, Lily Hoot from 3401, fellow CISDA members, Caleb Stanford, Alyssa Hwang, Daphne Ippolito, Paul He, Omar Navarro Leija and Jason Ma. I have enjoyed working with everyone towards bringing about changes in the PhD program, facilities and organizing events. CISDA has definitely been a big highlight of my time at Penn. I also thank others in the department who have supported me, especially Dan Roth, Linh Phan, Jianbo Shi and Nitish Gupta.

I want to thank Aditi Dudeja, Jiken Patel and Abhinav Rajvanshi who have been my closest friends over the last few years, celebrating birthdays and achievements together. I thank Gaurav Modi who was also my roommate for four years, and with whom me and my partner developed our interest in board games. I thank friends outside Penn who feel like they have been with me for a lifetime, Samarpita Patnaik, Arshia Garg, Sreejita Dutta, Nikhil Khicha, Mihir Pattani, Sarvesh Surana, Kanhaiya Kumawat, Devendra Sisodia and Sandeep RV. Whether we have met or not in the last few years, being in different cities or even continents, they continue to be an important part of my life and support system.

I thank my parents, Anil Agarwal and Archana Agarwal, for always supporting my education and career, regardless of wherever it led me. I thank my mom, the first PhD in the family, for being my first and best teacher. She is one of the most talented people I know. Until 6th grade, she read all my textbooks, made notes, simplified things for me and taught me

all of the school subjects. Without this, I would not have developed an interest in science and math. I thank my papa for spending so much time helping with my undergraduate applications, and for the long fun discussions on so many things. I know that if I want anything, he will move heaven and earth for it. I also thank my little brother, Kartik Agarwal. I love how close we have become as we have grown. He is a strong pillar of emotional support in my life. I am so proud of him and look forward to his move here in the next few months. I thank my parents-in-law, Nanalal Makwana and Taru Makwana, and my sisters-in-law, Rani Patel, Avni Vachhani and Binney Rojiwadia, for supporting me and celebrating me like my own parents and siblings. They are few of the nicest people I have ever known and I am glad that they are a part of my life.

The most important person in my life, someone who has been by my side supporting me throughout this program, is my family and partner, Raj Patel. I am so lucky to have met him and I cannot imagine life without him. He is a wonderful person who is smart, kind and humble. He cares tremendously about the people in his life. He makes me feel like I can do anything. These six years together have been the best of my life. This thesis belongs to him as much as it does to me. His love and support always keep me moving. I thank him for the joy in my life.

# ABSTRACT

## PRACTICAL NAMED ENTITY RECOGNITION: THE ROLE OF ENTITY AND ITS CONTEXT

Oshin Agarwal

Ani Nenkova

Neural supervised models for named entity recognition perform well within the same domain but fail to recognize entities not seen in the (pre-)training data with high accuracy. For better generalization, it is essential that models are able to recognize predictive contextual clues.

In this thesis, we explore the role of entity names and the context (sentence) in which they appear in named entity recognition. We quantify the generalization ability of models by probing them for the degree of learning names vs. contexts. We define constraining contexts as contexts with strong selectional preferences for the entity type. We argue that for constraining contexts, models should be able to recognize the entity type correctly regardless of the word identity. At the same time, we recognize that there is a generalization limit for named entity recognition based on the prevalence of constraining contexts, the accuracy of their automatic identification, and the names appearing in the model (pre-)training data for other contexts. We determine the feasibility of developing such a model by conducting human studies and by developing methods for the identification of constraining contexts.

From a practical perspective, since named entity recognition models are often developed for targeted applications, we also examine the robustness of models to challenges encountered in practice. Specifically, we study the effect of entities from different countries of origin, the effect of fine-grained topics within a domain often treated as homogeneous, and the effects of temporal changes. While it is challenging to identify the areas where model performance may suffer given the homogeneity of benchmark datasets, a practical solution for better performance remains to collect representative training data samples for each such area.

TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF ILLUSTRATIONS

# CHAPTER 1

# Introduction

Neural supervised models for several language processing tasks achieve impressive performance on benchmark datasets and are also fairly robust for practical applications. One such task is named entity recognition (NER), which involves recognizing entities and their semantic types in text. Entities can be anything ranging from the more commonly observed ones such as a person, an organization, or a location, to others such as an event (Hovy et al., 2006), a disease (Doğan et al., 2014), a gene (Kim et al., 2003), a chemical (Krallinger et al., 2015), a food item (Magnolini et al., 2019), an item of clothing (Putthividhya and Hu, 2011), a research technique (Augenstein et al., 2017a), etc. The recognized entities are used in both direct and indirect applications since named entities are often targets of interest in documents. Several other tasks such as summarization (Nobata et al., 2002; Zhou et al., 2004) and relation extraction (Hasegawa et al., 2004) target entities and can be supported by NER. Recognition of named entities has also been used to improve masking-based pre-training of language models (Guu et al., 2020) and even to develop improved evaluation methods for summarization (Eyal et al., 2019) and coreference resolution (Agarwal et al., 2019). Given the extensive practical use of NER, it is important to ask whether these models can generalize to diverse names and contexts.

While there is much prior work on the adaptation of models (Daumé III, 2007; Wang et al., 2020; Gururangan et al., 2020) to different genres and languages, there is also a need to carefully examine the in-domain robustness of NER along with how domains evolve, since practical NER models are often developed for in-domain applications. The overall performance of models evaluated on in-domain data is remarkably high but may not reflect the performance on under-represented or unseen samples. Prior work has shown that even within the same domain, there is a huge performance gap between entities seen and unseen

in the training data (Augenstein et al., 2017b) or the pre-training data (Ma and Hovy, 2016). We also study the in-domain robustness of NER models, focusing on several areas which will be encountered in practice.

We identify and study three different aspects where the performance of such a model may suffer in practice. Specifically, *we study the performance of models on ethnically diverse entities, the performance on sub-domains, and the performance over time.* We develop datasets, evaluation methodology and metrics, for the same.

**Ethnically diverse entities** We evaluate the robustness of NER models in recognizing diverse entities in different contexts, by comparing their performance across groups based on national origin. We create entity-switched datasets, replacing named entities in the original text with plausible named entities of the same type but of different national origin. An example is shown in Table 1. With such a replacement, we can measure the extent of recognition of the same entity in many contexts and many entities in the same context. We find that entities from certain origins are more reliably recognized than entities from elsewhere.

| Original Sentence | New Sentence |
|---|---|
| Defender Hassan Abbas PER rose to intercept a ... | Defender Ritwika Tomar PER rose to intercept a ... |
| The Democratic Convention signed an agreement on government and parliamentary support with its coalition partners the Social Democratic Union ORG and the Hungarian Democratic Union ORG. | The Democratic Convention signed an agreement on government and parliamentary support with its coalition partners the Jharkhand Mukti Morcha ORG and the Mizo National Front ORG. |

Table 1: Example of switching entities by national origin

**Sub-domains** We also evaluate the robustness of models on sub-domains within a seemingly homogenous domain. We collect a dataset of news articles from the New York Times and annotate it for named entities. We find that the performance of NER models varies significantly even in this dataset when it is stratified based on news topics. While entities unseen in the training data can be a factor that contributes to performance degradation, we

find that structural differences in sentences and insufficient contextual clues are the main contributors. Examples from different news topics are shown in Table 2.

| Domain | Sentence |
|---|---|
| Foreign | Mr. Rademaker noted, referring to a treaty with Russia. |
| Metropolitan | Mrs. Cohall, who attended public schools in Brooklyn, said she had always been a proponent of public education ... |
| Classified | WEISER–Joel, passed away on March 31st, 2007. |
| Sports | Pollin clashed with Jordan at a bargaining session during the long labor standoff in November 1998. |

Table 2: Example of sentences by sub-domains

**Temporal changes** Lastly, we discuss the impact of temporal changes on the performance of pre-trained models on downstream tasks. Keeping the performance of language technologies optimal as time passes is of great practical interest. Work in this area is very limited in NLP, especially on NER. We first examine the prior work to establish clearly the research questions we seek to ask and the appropriate experimental setup and evaluation method to answer them. We experiment with an existing NER dataset of tweets annotated for entities. We also work with datasets on other tasks that do not require manual annotation because annotating large NER datasets is laborious. The experiments reveal the distinction between temporal model deterioration and temporal domain adaptation becomes salient for systems built upon pre-trained representations. Models may or may not experience deterioration, though improving performance by adaptation through retraining is still possible.

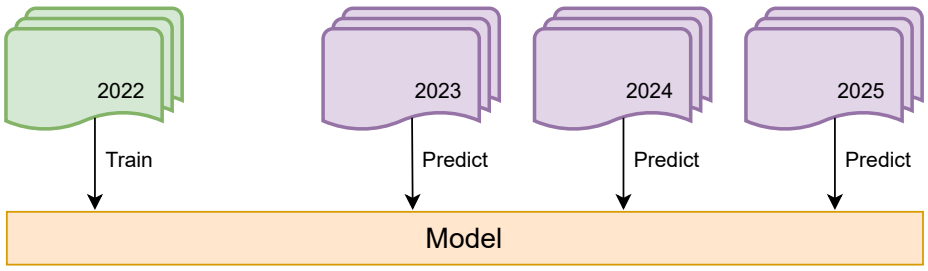



Figure 1: Model performance may vary over time

While identifying the areas where model performance may suffer is challenging, a straightforward solution to improve performance in the above areas is to simply collect more training data, representative of the samples with lower accuracy. However, an ideal model would rely less on memorizing entity names and be able to generalize to unseen entities. Consider the sentence "Dr. Smith is a computer scientist". Smith is a fairly common name, likely to be present in the (pre-)training data and would be easily identified by NER models. However, if instead of Smith, the sentence had an Indian name such as 'Patnaik', or a new popular name such as 'Khaleesi' [1], or even a hypothetical name such as 'Qwerty', we should still be able to recognize the entity based on the context because only a person may be a computer scientist.

Probing the reasons behind the model predictions i.e. whether the name or the context was learned, is needed to quantify whether models will generalize to unseen names and to develop robust models. By developing models that have access to only part of the input sentence, specifically the name or the context (Figure 2), in this thesis, *we examine not only the degree to which NER models use entity names to make predictions but also the degree to which they use entity contexts.* We found that while systems obtain high performance using just the word identity, the same is not true when just the context is used.

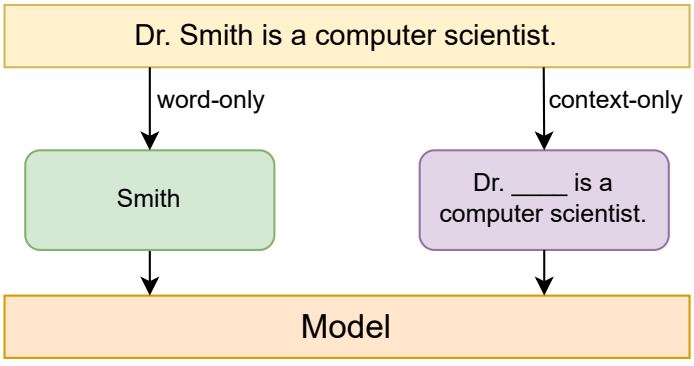



Figure 2: Probing name and context learning

Given the over-reliance on names and under-utilization of context, we propose the de-

[1] https://tinyurl.com/fw93ny78

sign of a model for NER. We define a constraining context as the sentence-level context of a word that has strong selectional preferences for the word type. We can determine the type of the word in such a context regardless of the word itself. A model should be able to determine if a context is constraining. If it is indeed constraining, as in the example above, the model prediction for any word in the context should only rely on the context. If the context is not constraining, word identity should be incorporated to make a prediction. In the latter case, we can not expect to recognize all entities correctly and the accuracy for entities in a non-constraining context would depend on the entities seen in the (pre-)training data. *We determine the feasibility of developing such a generalization limit-aware model by examining the strength of the selectional preferences imposed by entity contexts, through a human study.* We ask human annotators to determine the entity type solely on the context and use the distribution of labels to determine if the context is constraining or not. We find that the task of determining entity type based solely on the context is hard and a careful design of the study is crucial. As for the contexts, we found that roughly half of the contexts are constraining and not all are recognized by models.

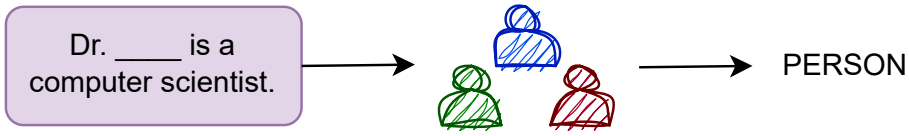



Figure 3: Human study for recognizing constraining contexts

We explore two methods (context-only model and MLM-kNN) to automatically recognize constraining contexts, eventually concluding that their automatic identification is hard. Even when we are able to recognize a constraining context, the dataset label might not agree with the context-based label in certain cases. Instead, these methods can be used to expand the set of plausible labels for contexts in the training data instead. We use the expanded set of plausible labels for a context to generate new examples by entity-switching and perform training data augmentation with them. We observe consistent improvements with the augmentation in low-resource settings.

## 1.1 Thesis Contribution

We examine the role of entity vs. its context in named entity recognition, specifically studying the feasibility of generalization based on the strength of the selectional preferences imposed on the entity types based on the sentential context. We argue that a NER model would identify constraining contexts i.e. strong selectional preferences and be able to identify any entity in such a context, and only rely on the entity identity otherwise. The prevalence of constraining contexts and the entities seen by models in the case of non-constraining contexts would then determine the limit on generalization in NER. However, we find that while constraining contexts are quite frequent, they are hard to identify automatically and the true entity label might still contradict the preference in certain scenarios. Therefore, from a practical perspective, the easiest solution for optimal performance is to incorporate representative samples of areas where the performance of the models suffers. Identifying these areas is challenging, given the overall high model accuracy on benchmark datasets. Studying practical tasks with benchmarks is also challenging since small design decisions can make the setup infeasible in practice or the results unreliable. We explore both these challenges in this thesis for the effect of entity, domain and temporal changes on robustness.

## 1.2 Thesis Outline

We first discuss the practical challenges in robustness. In Chapters 3, 4 & 5, we describe the work on entity-switched datasets, performance across sub-domains and temporal effects respectively. In the next two chapters, we discuss the predictiveness (or ambiguity) of the entity contexts and the feasibility of generalization based on it. In Chapter 6, we discuss our work on probing models for name vs. context learning. In Chapter 7, we describe a series of human evaluations in determining the predictiveness of contexts. We also present an approach to data augmentation for low-resource settings, based on the set of plausible entity labels in a context. In Chaper 8, we overview prior and potential future work on fine-grained NER including its extension to unseen entity types.

# CHAPTER 2

# Background and Related Work

*This chapter is based on content originally published in: Oshin Agarwal, Yinfei Yang, Byron C. Wallace and Ani Nenkova, 2020, Entity-switched datasets: An approach to auditing the in-domain robustness of named entity recognition models. arXiv preprint arXiv:2004.04123. (Agarwal et al., 2020), Oshin Agarwal, Yinfei Yang, Byron C. Wallace and Ani Nenkova, 2021, Interpretability analysis for named entity recognition to understand system predictions and how they can improve. Computational Linguistics, 47(1):117–140. (Agarwal et al., 2021b), and Oshin Agarwal and Ani Nenkova, 2023, The utility and interplay of gazetteers and entity segmentation for named entity recognition in English. In Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, Online. Association for Computational Linguistics. (Agarwal and Nenkova, 2021).*

## 2.1 Named Entity Recognition Models

Named Entity Recognition is formulated as a sequence labeling problem, where given a sequence of tokens $X = \{x_1, x_2, ....x_n\}$, the goal is to predict labels for each token, or $y = \{y_1, y_2, ....y_n\}$. X is generally the ordered sequence of words in a sentence and $y_i$ belongs to a pre-defined set of labels[2]. The set of labels is expanded based on a tagging scheme to denote the position of the word within an entity span. For example, under the IOB tagging scheme, each entity label $t$ is expanded to two labels—*B-t* and *I-t* where B denotes the first word of an entity and I denotes the remaining words. O is used for non-entity words. There are two conventional settings of NER – *i)* flat NER where each word can have only one label and only the longest entity comprising each word is annotated, and *ii)* nested NER

[2]labels, tags and classes are used interchangeably.

where each word can have multiple labels and all overlapping and nested entities are also annotated. For example, flat NER would include "New York City Department of Education" as an organization, whereas nested NER would also include "New York City" as a location. We work with the more commonly used setting of flat NER, which is also used by most works on robustness.

**Neural networks for NER** Neural networks were first used for NER as part of a unified architecture for many NLP tasks (Collobert and Weston, 2008; Collobert et al., 2011). A language model is first learned with unlabeled Wikipedia text, followed by finetuning the learned parameters jointly on a multitude of tasks such as parts-of-speech tagging, semantic role labeling and named entity recognition among others. The model generates a sentence representation for each input sentence using a stride-1 1D convolutional network, also known as a time delay neural network. Word representations are created by concatenating position vectors as the relative distance of each word from the target word to be classified. This is followed by a CRF-like layer for decoding but without normalization across all possible tags. A similar approach was adopted in subsequent models (Huang et al., 2015) using bidirectional LSTMs to generate token representations instead of a sentence-level representation approach. Bidirectional LSTMs concatenate the representation of the word conditioned on the left context (forward LSTM) and the representation conditioned on the right context (backward LSTM). A CRF is used as the final layer for decoding labels. The input is represented using pre-trained word embeddings, such as GloVe (Pennington et al., 2014a), SeNNA (Collobert et al., 2011) or word2vec (Mikolov et al., 2013).

**Character-level representation** The use of word-level vectors such as GloVe meant that any word not in the GloVe vocabulary would be treated as an unseen word and have the same input representation. To reduce the out-of-vocabulary words, character-based embeddings for words were incorporated along with the pre-trained word embeddings. All characters in the vocabulary are randomly initialized with fixed-length vectors which are updated during the training process. A word-level representation is generated using a biLSTM (Lample

et al., 2016) by concatenating the representation of the last letter using the forward LSTM and the first letter using the backward LSTM. Alternatively, it is generated using a CNN (Chiu and Nichols, 2016; Ma and Hovy, 2016) over the sequence of characters in a word with max pooling over time.

**Contextual representation** Embeddings such as GloVe learn a single vector for each word and therefore cannot handle polysemous words such as "bat" that could be an animal or a stick or the verb for hitting something. To overcome this, contextual representations were introduced that generate embeddings for words, given the context in which they appear which is generally the full sentence. ELMo (Peters et al., 2018) is a bidirectional language model for predicting the next word using CNN for character-based representations of words as input. It uses a two-layer LSTM and concatenates the output of both layers, finally taking a weighted linear combination of the original representation and the representation from the forward LSTM and the representation from the backward LSTM. A given sentence is passed through this network to generate a context-dependent representation for the words. BERT (Devlin et al., 2019) also generates contextual embeddings but by using a masked language modeling (MLM) objective. A word-piece (Wu et al., 2016) is randomly masked and the objective is to predict its original value using transformer encoders with self-attention to jointly condition on the left and right context. These representations have been shown to achieve state-of-the-art performance on a plethora of tasks.

Words are often described in the literature as the primary indicators for the entity type, with context serving as a discriminator when the word itself is ambiguous. A common example is the highly ambiguous word "Washington" which may be a person (George Washington), location (Washington DC), or even part of an organization (University of Washington), and the context helps identify the correct type. Context is never, to our knowledge, discussed as the primary indicator, with the word serving as the discriminator due to ambiguous context. The former view is more natural, intuitive and maybe even practical[3] since the word is such

[3] we explore this empirically in this thesis

a strong indicator for its type, so much so that a fairly good NER system can be obtained with simple word lookup as we show in Chapter 6. Regardless of the two views, in practice, most systems consider the full sentence with both the word and the context, without explicitly scoring the two components. This naturally leads to the models memorizing names which are strong indicators and poses a challenge in NER generalization, even within the same domain. We now discuss prior and contemporaneous related work that identified this memorization phenomenon, as well as the methods that incorporate clues based on either the word or the context explicitly as signals in the systems, though without considering their absolute or relative predictiveness.

## 2.2 Memorization of Entity Names

NER systems recognize entities seen in training more accurately than entities that were not present in the training set (Augenstein et al., 2017b; Fu et al., 2020b,a). Both traditional models with hand-crafted features (Finkel et al., 2005; Okazaki, 2007) and more recent neural network approaches (Collobert et al., 2011; Huang et al., 2015; Peters et al., 2018; Devlin et al., 2019) achieve lower performance on entities unseen in the training data.

Words in the training data are learned well, making it easier to recognize same or similar names that occur frequently even in the test set. The original CoNLL '03 NER shared task used a name look-up table as a baseline. Even though the simplest learning systems outperform such a baseline (Tjong Kim Sang and De Meulder, 2003), overviews of NER systems (Tjong Kim Sang and De Meulder, 2003; Yadav and Bethard, 2018) indicate that the most successful systems incorporate gazetteers (Kazama and Torisawa, 2007a; Ratinov and Roth, 2009) or lists of names of a given type. Even contemporary systems that do not use gazetteers expand their knowledge of names through the use of pre-trained word representations. With distributed representations trained on large background corpora, a name is "seen in training" if its representation is similar to names that explicitly appeared in the training data for NER. Consider, for example, the commonly used Brown cluster

features (Brown et al., 1992; Miller et al., 2004). Authors show examples of representations that would be the same for classes of words (John, George, James, Bob or John, Gerald, Phillip, Harold, respectively). If one of these names is seen in training, any of the other names can be considered as seen.

Similarly, using neural embeddings, words with representations similar to those seen explicitly in training would likely be treated as "seen" by the system as well. Tables 6 and 7 in Collobert et al. (2011) show the impact of word representations trained on small training data also annotated with entity types compared to those making use of large amounts of unlabeled text. When using only the limited data, the words with representations closest to *france* and *jesus* respectively are "persuade, faw, blackstock, giorgi" and "thickets, savary, sympathetic, jfk", which seem unlikely to be useful for the task of NER. For the word representations trained on Wikipedia and Reuters,[4] the corresponding most similar words are "austria, belgium, germany, italy" and "god, sati, christ, satan". These representations clearly have higher potential for the task of NER.

The extent to which learning systems are effectively 'better' look-up models — or if they actually learn to recognize contexts that suggest specific entity types — is not obvious. Our work in Chapter 6 on Name vs. Context Learning quantifies this.

## 2.3 Challenging Datasets and Generalization Metrics

Challenging datasets and new metrics have been developed to counteract the increasingly saturated performance on benchmark datasets where several entities in the datasets are same or similar to those in the training set. Broad Twitter Corpus (Derczynski et al., 2016) was intentionally created to consist of tweets with geographical, temporal and author diversity since the authors found that the most frequent names in CoNLL '03 are from the 1996 Pakistan cricket team. The authors of WNUT 2017 (Derczynski et al., 2017) take a more direct approach and forego the natural distribution of entities. They select text from social

[4]CoNLL data, one of the standard datasets used to evaluate NER systems, is drawn from Reuters.

media with emerging and rare entities and minimize the overlap between training and test set. This dataset can even be viewed as out-of-domain testing because even though the train and test genre is the same, the sources of text are different (Twitter, Reddit, Youtube).

In our literature survey on the use of gazetteers in NER (Agarwal and Nenkova, 2021), we provided statistics on the properties of both these datasets that impact robustness. Both BTC and WNUT had a large variety of distinct entity surface forms (60% & 88% respectively), much larger than CoNLL (50%). Both also have several lowercase entities. Only 55% and 28% entities are title cased in WNUT and BTC respectively as compared to 80% in CoNLL. BTC even exhibits a different distribution of capitalization between the training and the test set. In the training set, roughly half of the entities are sequences of words with capitalized first letters but this number is just 28% in the test set. The number of test entities seen in training is also less. This number is 0% for WNUT (by design), 41% in BTC and 62% in CoNLL. While most of the WNUT entities are seen in pre-training (GloVe+ELMo), only 65-75% of BTC entities are seen in these pre-training corpora, much smaller than other popular datasets.

Collecting such datasets from scratch is cumbersome. Therefore, perturbation techniques have been used in several recent works to create challenging and adversarial datasets either automatically or with humans in the loop. Prabhakaran et al. (2019) study the impact of person names on toxicity in online comments by substituting in names of controversial personalities. Emami et al. (2019) make coreference resolution robust to gender and number cues by making both antecedent and candidate of the same type. Subramanian and Roth (2019) use perturbation for adversarial training for coreference resolution of person and location entities. Raiman and Miller (2017) augment question-answering datasets via perturbations and Kaushik et al. (2019) create manually perturbed examples by experts for sentiment analysis and natural language inference. However, there is no large-scale dataset available for the evaluation of NER. We perform such an evaluation in Chapter 3 on Entity Effects.

Besides creating challenging datasets, prior work has measured generalization by breaking down model performance by dataset properties. Starting Augenstein et al. (2017b), the performance of entities seen and unseen in training data is occasionally reported separately since unseen entities are harder to detect. Ma and Hovy (2016) extended this by reporting performance on entities seen in both training and pre-training, those that are seen in one but not the other and those that are not seen in either. Reporting such a breakdown is now difficult with fine-grained tokenization, finetuned models instead of feature-based ones and contextual representation that provide an initial representation impacted both by the pretraining data and the given context.

Fu et al. (2020b) presents a more nuanced breakdown of performance by determining the ambiguity of test entities as the similarity of their tag distribution in the test and training set. Such multi-statistic reporting while informative can be cumbersome. Derczynski et al. (2017) instead introduce the surface form F1 which counts each surface form in the test set only once so that the metric reflects the ability to recognize diverse entities and not just frequent ones. Our intent with the work in Chapter 3 is similar but we perform entity replacement since entities in a dataset are often repetitive or similar and we want to measure the recognition of entities in diverse contexts.

## 2.4 Incorporating Entity Gazetteers

Memorization of entities remains a concern in measuring generalization in NER. Despite "anti-memorization" efforts by developing several datasets and metrics, a good practical way to boost performance on them is still to increase entity coverage during training. Overviews of named entity recognition systems indicate that the most successful systems, both old (Tjong Kim Sang and De Meulder, 2003) and recent (Yadav and Bethard, 2018), make use of gazetteers.

Gazetteers are large dictionaries consisting of lists of entities of a particular type. For example, a person gazetteer may consist of full names and parts of names such as the first

names of people. Existing tables, lists, directories, databases and knowledge bases are widely available and can be used to derive them. Some researchers have specifically compiled various resources to form gazetteers, while others make use of those provided in prior work. Early work collected gazetteers from the CIA factbook for geographic locations, lists of popular person names, etc (Mikheev et al., 1999). More recently, Ratinov and Roth (2009) derived a gazetteer from the Web and Wikipedia and Chiu and Nichols (2016) used DBPedia.

Here, we overview methods to incorporate gazetteers in NER models.

### 2.4.1 Discrete Gazetteer Lookup Features

Feature-based CRF models for NER used gazetteers to generate indicators for each word in a sentence (Bender et al., 2003; Minkov et al., 2005; Ratinov and Roth, 2009; Ritter et al., 2011; Yang et al., 2016; Seyler et al., 2018). The number of indicators equals the number of entity types in the dataset and indicate (with a binary 1/0 value) if the word is part of a gazetteer entry of the given type.

Many neural network approaches continue to incorporate gazetteers as discrete indicator features concatenated to the pre-trained word embeddings as the input (Collobert et al., 2011; Huang et al., 2015). Adding the features in later stages does not work as well (Magnolini et al., 2019). Both Collobert et al. (2011) and Huang et al. (2015) pre-process datasets to match the gazetteer entries to sentences, using both exact matches and multi-word partial matches to gazetteer entries. Chiu and Nichols (2016) perform a similar matching but use four binary values for each label, indicating whether the given word matches the gazetteer entity exactly (S), at the beginning (B), end (E) or any of the words in between (I).

### 2.4.2 Continuous Gazetteer Features

The approach above does not use gazetteers very effectively. Gazetteers contain many more entities of each type than are available in even the largest training set. One insight is to use the gazetteers as an additional source of training examples. A simple way is to add the

gazetteer entries to the labeled data, without any context. Liu et al. (2019a) report that this data augmentation approach led to much worse overall results, presumably because of the great shift in label distributions. Another approach is to augment the training data by replacing entities in place with other entities from gazetteers. Song et al. (2020) reported no improvement with such a random entity replacement, likely due to the need for manual intervention for the replacement of entities of some types to maintain the coherence of text (Agarwal et al., 2020).

A much more successful alternative is to learn a separate (or sub-) module, trained to predict types for text spans, using the gazetteer entries and synthetic negative examples sampled from a NER training set or even the gazetteer. We will refer to the separate module as a *gazetteer network*. It is straightforward to integrate the label distribution scores from this model in a semi-Markov CRF for sequence labeling (Ye and Ling, 2018) that operates at the span level (which we describe in greater detail later). The resulting combination is far more effective than discrete indicator gazetteer features.

Magnolini et al. (2019) and Liu et al. (2019c) propose a similar approach. They learn a gazetteer network but instead of using the label score distribution, intermediate word representations (*gazetteer embeddings*) are incorporated in the NER model. Liu et al. (2019c) use a semi-Markov CRF operating at the span level and generate the gazetteer embeddings for each potential span. They follow the evaluation approach of Ma and Hovy (2016), breaking down results by whether an entity was seen only in training, seen only in pre-training, seen in both and seen in neither. The largest improvement was in the "seen in neither" subset, showing that this approach is particularly helpful for out-of-vocabulary words with respect to the training and pre-training data.

Magnolini et al. (2019) use the standard word-level CRF and hence do not have spans available so they input the full sentence to the gazetteer network. This makes the training and inference setup for the gazetteer network different as entity phrases are used as input during training. They reported mixed results for this approach. In our experiments, we

evaluated their method on a larger number of datasets but used a different approach for negative sampling for the gazetteer network training data. We observed some improvement on almost all datasets, contingent on the input representation.

### 2.4.3 Contextual Gazetteers

Learning from just the gazetteer has the drawback that the representations do not include any clues about the context in which the entity types are used. The same surface form may appear in multiple entity gazetteers with different types. Given that current methods heavily rely on entity memorization and little on context, this is possibly acceptable. For completeness, however, we ought to mention that the link structure of Wikipedia can be used to derive dense representations for entity types directly (Long et al., 2016; Ganea and Hofmann, 2017; Mengge et al., 2020; Ghaddar and Langlais, 2018b). Similarly, masked language models can be used to generate alternate entities for sentences in existing datasets to create additional training data (Zhou et al., 2022; Ding et al., 2020).

We reproduced several of these methods in identical settings (Agarwal and Nenkova, 2021) and found the simplest method i.e., discrete word-matching based gazetteer feature vector, shows the most consistent improvement across datasets and representations.

Our work in Chapter 7 on data augmentation indirectly improves the coverage of both entities and contexts by creating new examples via the mixing and matching of entities and contexts within the training data. No new entities from any databases are introduced. Instead, other entities appearing in the training set are used. These entities are either of the same type as the original entity or are of other entity types plausible in the original context.

## 2.5 Incorporating Context

Although most works on NER have focused on learning to recognize entities using the full sentence, without distinguishing the context from the word and methods to improve entity coverage remain popular, some early work on NER did explicitly deal with the task of scoring

contexts on their ability to predict the entity types in that context.

Approaches for database completion and information extraction use unannotated text to learn patterns predictive of entity types (Riloff and Jones, 1999; Collins and Singer, 1999; Agichtein and Gravano, 2000; Etzioni et al., 2005; Banko et al., 2007) and then use these to find instances of new names. Given a set of known names, they rank all $n$-gram contexts for their ability to predict the type of entities, discovering for example that "the mayor of X" or "Mr. Y" or "permanent resident of Z" are predictive of city, person, and country respectively. Early NER systems also use additional unannotated data to identify predictive contexts. These however had little to no effect on system performance (Tjong Kim Sang and De Meulder, 2003) with few exceptions where both names and contexts were bootstrapped (Cucerzan and Yarowsky, 2002; Nadeau et al., 2006; Talukdar et al., 2006).

Recent work in NLP relies on pre-trained representations to expand the ability of the systems to learn context and names (Huang et al., 2015). So far, there has not been an analysis of which parts of contexts are properly captured by the representations, especially what they do better than more local representations of just the preceding/following word. Moreover, contextual representation (Peters et al., 2018; Devlin et al., 2019) make such an interpretation difficult since the context is incorporated even in the input representation. We perform such an analysis in Chapter 6.

There are a few very recent exceptions that analyse entity contexts or even incorporate them in models explicitly. Fu et al. (2020b) use dense $n$-gram representations to compare the similarity of train and test entity contexts. They find that context similarity does influence performance but not nearly as much as entity overlap. They do not analyse the predictiveness or the ambiguity of the context.

Lin et al. (2020) collect human judgments of which words in the context support the identification of the entity by given type. To avoid any influence due to the word itself, they replace it with its type when presenting the sentences to humans. They then incorporate

these words as additional supervision during training and found that this reduced the amount of data necessary to obtain comparable performance. By essentially removing the words not needed for entity recognition, the task is made easier with these signals. It is unclear whether they found that there were instances where humans did not find any context word appropriate for the given type, or if annotators found the same words equally predictive for another type besides the given one. We perform a human evaluation in our work in Chapter 7 to determine which contexts are predictive. We treat the context as a whole and do not aim to find specific relevant words.

Most closely related to our work in Chapters 6 & 7 is a follow-up work (Ghaddar et al., 2021) that develops a dataset where contextual clues are sufficient to predict entity types. Like us, they also build a context-only tagger but by replacing each word in a dataset with its label from a NER model and training an $n$-gram language model on it. The perplexity of each label in a given context is then computed to predict the type with the word itself. They use a label confidence cutoff to identify predictive contexts. Specifically, they consider that the context has obvious clues to determine the entity type is a weak tagger using the full sentence as input predicts an incorrect label with high confidence by the context-only tagger predicts the correct label with a high confidence gap between the two most probable labels. Using this method, they develop a dataset where contextual clues are sufficient to predict the entity type. Unfortunately, the data was not public at the time of writing this thesis and thus, we could not compare our work with it.

# CHAPTER 3

# Entity Effects

*This chapter is based on content originally in: Oshin Agarwal, Yinfei Yang, Byron C. Wallace and Ani Nenkova, 2020, Entity-switched datasets: An approach to auditing the in-domain robustness of named entity recognition models. arXiv preprint arXiv:2004.04123. (Agarwal et al., 2020). In this chapter, we evaluate the robustness of Named Entity Recognition models in recognizing entities from different countries of origin by creating entity-switched datasets.*

## 3.1 Introduction

Named entity recognition (NER) systems perform well on standard datasets comprising English news. But given the paucity of data, it is difficult to draw conclusions about the robustness of systems with respect to recognizing a diverse set of entities. Research in other areas of predictive technology has revealed that ostensibly strong predictive performance may obscure wide variations in performance for certain types of data. For example, gender recognition systems attain high accuracy on what used to be considered standard datasets for this task, but have large error rates on people with dark skin tone, particularly on women with dark skin (Buolamwini and Gebru, 2018). Language identification is also highly accurate on standard datasets (Zhang et al., 2018) but may fail to recognize dialects, e.g., failing to identify African American English as English (Blodgett et al., 2016). Here, we set out to develop methods for testing two intertwined properties of NER models: (i) Their robustness on a variety of entities, and, (ii) their relative performance across groups, which here correspond to national origin. It is known that the robustness of NER methods depends on entities being represented in the training data (Augenstein et al., 2017b). Probing their

| Original Sentence | New Sentence |
|---|---|
| Defender Hassan Abbas PER rose to intercept a ... | Defender Ritwika Tomar PER rose to intercept a ... |
| The Democratic Convention signed an agreement on government and parliamentary support with its coalition partners the Social Democratic Union ORG and the Hungarian Democratic Union ORG. | The Democratic Convention signed an agreement on government and parliamentary support with its coalition partners the Jharkhand Mukti Morcha ORG and the Mizo National Front ORG. |

Table 3: Example of switching entities by national origin

performance through the lens of national origin is then one way to question the choice of training data, and what a system can learn from it.

We propose a method for auditing the in-domain robustness of systems, focusing on differences in performance due to the national origin of entities. We create *entity-switched* datasets (Agarwal et al., 2020), in which named entities in the original text are replaced by plausible named entities of the same type while retaining the rest of the text and maintaining its coherence. An example is shown in Table 3 where the original entities are replaced by Indian ones. Entity-switched datasets not only have a diverse set of entities but also the same entity in a variety of contexts by replacing all entities of a given type with the same entity throughout the dataset. They can be used to perform a large-scale evaluation of NER for robustness using existing datasets without the need for manual labeling of more data.

We evaluate state-of-the-art systems on these *entity-switched* datasets and find that they have the highest performance (F1) on American and Indian entities and the lowest performance on Vietnamese and Indonesian entities. The datasets are available at https://github.com/oagarwal/entity-switched-ner.

## 3.2 Replacing PER Entities

In the first group of datasets, we change only the names of people. We replace all *PER* entities in the test set with the same string. For example, all sequences of *PER* might be replaced with the name 'John Smith'. Names for replacement are drawn from lists of

popular names in the countries with the largest populations. This allows us to examine system performance with respect to the country of origin, and also in terms of the number of people whose names would be potentially affected by recognition failures. Specifically, we selected the 15 most populous countries and found 20 common first names and 10 common family names for each[5]. We used two sets of Chinese names, from mainland China and Taiwan, respectively. We also use two sets of U.S. names: The first comprising common names (e.g., John Smith) and the second composed of Native American, African American, and names that could be locations (e.g., Madison) or regular non-entity words (e.g., Brown).

The first names used in the experiment are a mix of male, female, and unisex names. We create full names by matching each first name to a random family name. For some countries, names have additional constraints that we account for. In Indonesia[6], names might include only a single name or multiple first names (without a family name). In Pakistan[7], some female names have a first name followed by the father/husband's most called name.

We replace all *PER* spans in the test set with a single name. Therefore, $n$ versions of the test set are created where $n$ is the number of new names selected. Training data is unchanged, as our goal is to quantify the robustness of identifying various names. We attempt to be consistent in the replacement, i.e., full names are replaced by full names, first names by first names, and last names by last names. We treat space-separated multi-word names as full names. We take a Western-centric view and consider the first word to be a first name and the remaining to be the last name, and determine if other occurrences are first or last names by string matching. If a multi-word name is part of a longer name, we do not break it down and replace it based on the longer name it matches.

[5]Sources include: Wikipedia, websites with baby names, and websites listing popular names
[6]https://en.wikipedia.org/wiki/Indonesian_names
[7]https://en.wikipedia.org/wiki/Pakistani_name

## 3.3 Replacing Other Entity Types

We construct two more datasets in which we replace all *PER*, *LOC* and *ORG* instances with entities of the same type and from a particular country. We do not replace *MISC* entities, because these are not usually country-specific. In one dataset, we replace all entities with corresponding entities of Indian origin; in the other, with entities of Vietnamese origin. Unlike in the above dataset, where we replaced every entity of type *PER* with the *same* name throughout, we perform a stochastic (though type-constrained) mapping from the original set of entities to the new entity set. That is, when replacing a target entity $e$ of type $t$, we sample an entity with which to replace it at i.i.d. random from entities of type $t$ in the set of country-specific entities.

We select *PER* names as in the previous section. For each document, we then generate a list of possible names an entity was referred to by string matching and replace each entity with a new name as consistently as possible, as in the previous section. For *LOC*, we select names of villages, cities, and provinces and again select a location of the same type from the country-specific list in a document.

Consistently replacing *ORG* entities is more complicated than replacing *PER* and *LOC* entities because not every organization would be suitable for every context. We cannot, for instance, reasonably replace 'Bank of America' in 'We withdrew money from the Bank of America' with 'New York Times' or 'Mayo Clinic'. Therefore, we divided organizations into sub-categories: Airline, Bank, Corporation, Newspaper, Political Party, Restaurant, Sports Team, Sports Union, University, and Others. Others included international or intergovernmental organizations such as United Nations and we did not replace these. We selected candidates for each category from similar websites as above, labeled all test *ORG* instances with the sub-category manually, and then replaced them with a country-specific entity of the same sub-category, again being consistent within a document based on string matching.

| | GloVe words | | | GloVe words+chars | | | BERT subwords | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| *Original testset* | 96.9 | 96.5 | 96.7 | 97.1 | 98.1 | 97.6 | 98.3 | 98.1 | 98.2 |
| *Super recall* | | | | | | | | | |
| US | 96.9 | 99.6 | 98.2 | 96.9 | 99.6 | 98.3 | 98.4 | 99.7 | 99.1 |
| Russia | 96.8 | 99.5 | 98.1 | 97.1 | 99.8 | 98.4 | 98.4 | 99.3 | 98.9 |
| India | 96.5 | 99.5 | 98.0 | 97.1 | 99.3 | 98.2 | 98.4 | 98.8 | 98.6 |
| Mexico | 96.7 | 98.9 | 97.8 | 97.1 | 98.9 | 98.0 | 98.4 | 99.2 | 98.8 |
| *Poor recall* | | | | | | | | | |
| China-Taiwan | 95.4 | 93.2 | 93.9 | 97.0 | 94.9 | 95.6 | 98.3 | 92.0 | 94.8 |
| US (Difficult) | 95.9 | 87.4 | 90.2 | 96.6 | 87.9 | 90.7 | 98.1 | 88.5 | 92.3 |
| Indonesia | 95.3 | 84.6 | 88.7 | 96.5 | 91.0 | 93.3 | 97.8 | 85.8 | 92.0 |
| Vietnam | 94.6 | 78.2 | 84.2 | 96.0 | 78.5 | 84.5 | 98.0 | 84.2 | 89.8 |
| *BERT not best* | | | | | | | | | |
| Ethiopia | 96.5 | 96.8 | 96.6 | 96.6 | 98.6 | 97.9 | 98.3 | 90.6 | 94.1 |
| Nigeria | 96.3 | 92.2 | 94.1 | 97.1 | 96.6 | 96.8 | 98.2 | 90.2 | 93.8 |
| Philippines | 97.3 | 97.9 | 97.5 | 97.5 | 98.9 | 98.2 | 98.6 | 94.7 | 96.4 |
| *Other* | | | | | | | | | |
| Bangladesh | 96.7 | 97.5 | 97.1 | 97.1 | 97.6 | 97.3 | 98.4 | 97.8 | 98.0 |
| Brazil | 96.6 | 96.8 | 96.6 | 97.1 | 96.2 | 96.5 | 98.4 | 96.7 | 97.5 |
| China-Mainland | 95.7 | 97.9 | 96.7 | 97.0 | 97.4 | 97.2 | 98.4 | 96.7 | 97.5 |
| Egypt | 96.6 | 99.2 | 97.8 | 97.0 | 98.2 | 97.6 | 98.4 | 97.4 | 97.9 |
| Japan | 96.7 | 97.2 | 96.8 | 97.0 | 98.7 | 97.8 | 98.5 | 99.0 | 98.7 |
| Pakistan | 96.2 | 92.6 | 94.1 | 97.0 | 96.5 | 96.6 | 98.3 | 95.3 | 96.7 |

Table 4: Token-level F1 of PER entities in CoNLL '03 entity-switched test data. Original refers to the unchanged data. The rest of the rows are averaged over 20 names for each country.

## 3.4 Robustness Evaluation

We use the English CoNLL'03 (Tjong Kim Sang and De Meulder, 2003) with IO labeling and report the token-level micro-F1 for each country, replacing every *PER* entity in the test set with each of the 20 names in turn and taking the average over the 20 versions of the dataset. We evaluate the word-based biLSTM-CRF (Huang et al., 2015), word and character-based biLSTM-CRF (Lample et al., 2016) and BERT (Devlin et al., 2019). For the first two, we used 300-d cased GloVe (Pennington et al., 2014b) vectors trained on Common Crawl. For BERT, we use the public large uncased[8] model and apply the default fine-tuning strategy.

[8]Uncased performed better than cased.

| | GloVe words | | | GloVe words+chars | | | BERT subwords | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| Original | 94.7 | 95.6 | 95.2 | 97.5 | 95.0 | 96.2 | 97.0 | 96.8 | 96.9 |
| India | 94.2 | 95.5 | 94.8 | 97.0 | 95.7 | 96.2 | 96.3 | 96.9 | 96.6 |
| Vietnam | 93.1 | 82.3 | 85.8 | 96.3 | 82.3 | 86.9 | 96.5 | 85.2 | 90.5 |

Table 5: Token-level F1 of PER entities in OntoNotes newswire entity-switched test data. Original refers to the unchanged data. The rest of the rows are averaged over 20 names for each country.

| | GloVe words | | | GloVe words+chars | | | BERT subwords | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| Original | 90.9 | 91.4 | 91.2 | 90.9 | 92.6 | 91.7 | 95.5 | 93.3 | 94.4 |
| India | 84.3 | 77.3 | 80.7 | 83.8 | 82.9 | 83.3 | 95.6 | 87.8 | 91.5 |
| Vietnam | 74.3 | 73.0 | 73.6 | 77.9 | 76.4 | 77.2 | 96.2 | 81.5 | 88.2 |

Table 6: Token-level F1 of all entity types in the CoNLL '03 entity-switched test data.

We present the results in Table 4. All the models achieve higher F1 on common American names, Russian, Indian and Mexican names than the original dataset (*Super recall*). Precision remains the same but recall improves to almost perfect. For the GloVe models, performance drops by up to ∼10 points F1 for certain countries (*Poor recall*). Names from Indonesia and Vietnam fare the worst, along with the less common and difficult US names, and names from Taiwan, with small degradation of precision and a precipitous drop of recall. BERT exhibits a similar pattern, with stable precision and varying recall which remains above 84% for all name origins. Notably, BERT performance is lowest on names from Ethiopia, Nigeria, and the Philippines (see *BERT not best* rows). The lower performance can be attributed to both the coverage in the training data and greater ambiguity in names from certain countries of origin. In light of these findings, one might wonder if accepting current architectures trained on standard corpora as state-of-the-art is the NER equivalent of developments in photography, which was optimized for perfect exposure of white skin, and which is the assumed technical reason for many failures of computer vision applications when applied to dark-skinned people (Benjamin, 2019). At the very least practitioners should be cognizant of these systematic performance differences.

| | | | | | |
|---|---|---|---|---|---|
| Original | 96.5 | Egypt | 97.2 | Ethiopia | 95.4 |
| Russia | 98.1 | India | 97.3 | China-Taiwan | 94.3 |
| US | 98.0 | China-Mainland | 96.1 | Us-difficult | 94.4 |
| Mexico | 97.8 | Pakistan | 96.9 | Nigeria | 91.3 |
| Japan | 97.7 | Philippines | 96.6 | Indonesia | 89.3 |
| Bangladesh | 97.6 | Brazil | 96.3 | Vietnam | 87.3 |

Table 7: Span-level F1 of PER entities in CoNLL '03 entity-switched test data. Original refers to the unchanged data. The rest of the rows are averaged over 20 names for each country.

| | |
|---|---|
| Original | 97.1 |
| India | 89.3 |
| Vietnam | 87.5 |

Table 8: Span-level F1 of PER entities in OntoNotes newswire entity-switched test data.

| | |
|---|---|
| Original | 91.8 |
| India | 82.6 |
| Vietnam | 77.1 |

Table 9: Span-level F1 of all entity types in the CoNLL '03 entity-switched test data.

BERT and character biLSTM-CRF results are higher and more stable, but it is nevertheless clear that one need not perform a completely out-of-domain test to observe deteriorating performance; changing the name strings is sufficient. We perform similar tests on the newswire section of OntoNotes (Pradhan and Xue, 2009). We use the original train and test splits, where we have switched *PER* in the latter. We use names from India and Vietnam because these were in the top and bottom-performing entity-switched sets, respectively, for CoNLL data. We observe a similar drop in performance (Table 5).

For the replacement of all entities, we observe a drop in performance on both datasets (Table 6). Both precision and recall drop for the GloVe systems. However, for BERT, the precision remains the same and only the recall drops.

For consistency with models and metrics used in all the later chapters, we perform these experiments again with bert-large-cased and report the span level entity F1. The results are shown in Tables 7, 9 and 8. The overall conclusion remains the same.

| True labels | Original | Indian | Vietnamese |
|---|---|---|---|
| Japan LOC began the defence of their Asian Cup MISC title with a lucky 2-1 win against Syria LOC in a Group C championship match on Friday. | Japan LOC began the defence of their Asian Cup MISC title with a lucky 2-1 win against Syria LOC in a Group C championship match on Friday. | Dhanbad LOC began the defence of their Asian Cup MISC title with a lucky 2-1 win against Thungapuram PER in a Group C ... | Long An O began the defence of their Asian Cup MISC title with a lucky 2-1 win against Bac Lieu PER in a Group C championship .... |
| Nader Jokhadar PER had given Syria LOC the lead with a well-struck header in the seventh minute. | Nader Jokhadar PER had given Syria LOC the lead with a well-struck header in the seventh minute. | Priya Khemka LOC had given Thungapuram O the lead with a well-struck header in the .... | Thien Hue LOC had given Bac Lieu PER the lead with a well-struck header in the seventh minute. |
| ROME LOC 1996-12-06 | ROME LOC 1996-12-06 | Thevaiyur O 1996-12-06 | Ha Long Bay 1996 -12-06 LOC |
| SOCCER - FIFA ORG BOSS HAVELANGE PER STANDS BY WEAH PER. | SOCCER - FIFA ORG BOSS HAVELANGE PER STANDS BY WEAH O. | SOCCER - Judo Federation ORG of O India LOC BOSS Dheeraj Uniyal PER STANDS BY Anjali Lal PER. | SOCCER - Vietnam Football Federation LOC BOSS Thu O Thai MISC STANDS BY Duc Tan PER. |
| In an interview with the Italian MISC newspaper Gazzetta dello Sport ORG, he was quoted as saying Weah PER had been provoked into the assault which left Costa PER with a broken nose. | In an interview with the Italian MISC newspaper Gazzetta dello Sport ORG, he was quoted as saying Weah PER had been provoked into the assault which left Costa LOC with a broken nose. | In an interview with the Italian MISC newspaper Hari Bhoomi PER, he was quoted as saying Masih PER had been provoked into the assault which left Damerla O with a broken nose. | In an interview with the Italian MISC newspaper Tien Phong PER, he was quoted as saying Hue PER had been provoked into the assault which left Giang LOC with a broken nose. |
| Cagliari ORG (16) v Reggiana ORG (18) 1530 | Cagliari ORG (16) v Reggiana ORG (18) 1530 | Thiruvanantha -puram LOC (16) v Hyderabad LOC (18) 1530 | Sanna Khanh Hoa Futsal Club PER (16) v Ha Long Bay ORG (18) 1530 |
| Bottom team Reggiana ORG are also without a suspended defender, German MISC Dietmar Beiersdorfer PER. | Bottom team Reggiana ORG are also without a suspended defender, German MISC Dietmar Beiersdorfer PER. | Bottom team Hyderabad LOC are also without a suspended defender, German MISC Navneet Nedungadi PER. | Bottom team Ha Long Bay PER are also without a suspended defender, German MISC Linh On PER. |

Table 10: Predictions of Huang et al. (2015) on the original, Indian and Vietnamese test sets. Many errors are made on the entity-switched datasets, even in the presence of strong contextual clues, including patterns common in the training data.

Some examples of replacement and predictions by the GloVe biLSTM-CRF are shown in Table 10. Models ignore predictive contexts and assign labels based on the word identity. In the first example, both the entities should be of the same type – *LOC* or *PER*; however for Indian entities, one is predicted as *PER* and the other as *LOC*; for Vietnamese, one is predicted as *PER* and the other isn't even recognized as an entity. The second example is indeed hard and a known word identity would be easier to detect. The third example has a common pattern '*LOC DATE*' from the training set but is not recognized for the switched entities. For Vietnamese, it even causes the date to be mispredicted as *LOC*. Similarly, the sixth example has another common pattern in the training data that the model fails to recognize. In the fifth and seventh examples, anything following 'newspaper' and 'team' are their respective names. Both should be *ORG* based on clear contextual clues but are still misrecognized. While it may appear the 'Italian newspaper' followed by a name from a different origin may be the cause of the error, in the last row all the names followed by German are recognized correctly irrespective of origin.

## 3.5 Limitations

Our datasets are limited to the top few populous countries. However, they can easily be created for more rare entities as well. One could even use a large gazetteer of entities to perform a more comprehensive evaluation. Our work is limited to coarse-grained common entity types. Extending to finer-grained entity types and entities in specialized domains such as biomedical or financial domains remains to be explored in the future. This limitation is mainly due to the need for manual intervention to sub-categorize entities of certain types for a coherent replacement.

## 3.6 Conclusion

Standard NER datasets such as English news contain a limited number of unique entities, with many of them occurring in both the train and the test sets. As a result, models

do not recognize 'foreign' entity instances as well. We introduced a practical approach of *entity-switching* to create datasets to test the in-domain robustness of systems. We selected entities from different countries and showed that models perform extremely well on entities from certain countries, but not as well on others. This finding has fairness implications when NER is used in practice.

# CHAPTER 4

# Domain Effects

*This chapter is based on content originally published in: Oshin Agarwal and Ani Nenkova, 2023, Named entity recognition in a very homogeneous domain. In Findings of the Association for Computational Linguistics: EACL 2023, Online. Association for Computational Linguistics. (Agarwal and Nenkova, 2023). In this chapter, we evaluate the robustness of Named Entity Recognition models across topics within a typically "homogeneous" domain, by collecting a new dataset of news articles stratified by news topics.*

## 4.1 Introduction

Supervised neural models for named entity recognition achieve high accuracy when used in-domain. When models are evaluated or adapted (Daumé III, 2007; Wang et al., 2020; Gururangan et al., 2020) for out-of-domain text, or even developed for specialized domains (Nguyen et al., 2020; Beltagy et al., 2019), the term domain generally refers to broad genres such as news, social media, or biomedical text. However, text can be (dis)similar in aspects beyond genre, such as the source of the data, its structure, or the time period. Dai et al. (2019) distinguish two aspects of domain—the genre and the tenor, which they describe as the participants in the discourse, their relationships and their purpose. They find that even though people consider genre to be more important for domain adaptation, tenor is important as well when selecting pre-training data.

The term domain encompasses more than just broadly defined genres. Online comments on different platforms can be considered different domains. So can news from different newspapers or different time periods. We show that even text from the same genre and

source needs to be examined finely for topical or structural differences. We collect a dataset of news articles from the New York Times and annotate it for named entities. We find that the performance of NER models varies significantly even in this dataset when it is stratified based on news topics. While entities unseen in the training data can be a factor that contributes to performance degradation, we find that structural differences in sentences and entity ambiguity are the main contributors. Selecting diverse data is therefore crucial even in such "in-domain" settings. We show that even a very small number of sentences from each topic can help narrow the performance gap, and selecting random sentences rather than full documents from the full corpus, will ensure that there is a good sample of diverse sentences.

## 4.2 Dataset

The dataset is available at https://github.com/oagarwal/nyt-ner. Here we describe the process of collecting it.

### 4.2.1 Data Collection

We sample sentences from the New York Times (NYT) Annotated Corpus (Sandhaus, 2008). The corpus consists of 1.8M articles from NYT between 1987 and 2007 along with article metadata provided by the New York Times Newsroom, the New York Times Indexing Service and the online production staff at nytimes.com. We select sentences from different years and news topics[9], both available as metadata. Variations in topic names are merged together resulting in a total of nine topics—Arts (+Weekend/Cultural), Business (+Financial), Classifieds (+Obituary), Editorial, Foreign, Metropolitan, Sports and Others. Others consists of all desks that did not have many articles such as Real Estate, New Jersey Weekly, Book Review, Job Market, Science and Health & Fitness.

[9]desk in NYT newsroom that produced the article

| | # sentences | | | # entities | | |
|---|---|---|---|---|---|---|
| | train | dev | test | train | dev | test |
| arts | 3570 | 1531 | 5101 | 2451 | 1112 | 3542 |
| business | 2454 | 1052 | 3507 | 2055 | 870 | 2923 |
| classified | 1052 | 451 | 1503 | 1380 | 568 | 1895 |
| editorial | 2872 | 1232 | 4104 | 2198 | 939 | 3113 |
| foreign | 4654 | 1995 | 6649 | 3961 | 1672 | 5906 |
| metropolitan | 2873 | 1232 | 4106 | 2254 | 888 | 3141 |
| national | 3888 | 1667 | 5555 | 3062 | 1310 | 4303 |
| sports | 3664 | 1571 | 5235 | 3475 | 1572 | 4995 |
| others | 3221 | 1380 | 4602 | 2397 | 988 | 3413 |

Table 11: NYT dataset statistics by topic

### 4.2.2 Data Annotation

The selected sentences are labeled with person (*PER*), location (*LOC*) and organization (*ORG*) tags on Upwork[10], with CoNLL'03 (Tjong Kim Sang and De Meulder, 2003) guidelines and annotation scheme. For efficiency, we first annotate the sentences with entities from the article metadata. The metadata consists of relevant persons, locations and organizations selected from a fixed vocabulary, manually assigned as part of NYT indexing. This first pass of annotation is done using phrase matching, similar to a gazetteer lookup. The resulting annotations are expected to be better than looking up in a general gazetteer since the available entities are assigned manually per article.

We use one annotator per example, but the annotators are first trained for the task. Each annotator is given 10-20 sentences to correct the entity labels from the first pass. The corrected sentences are reviewed by one of the authors and the feedback is shared with the annotator. Another 10-20 sentences are then shared with the annotator. These sentences are a mix of previously annotated but problematic sentences and new sentences, focusing on the types of mistakes made by the particular annotator in the earlier batch. If the annotator makes several mistakes in this round overall, or even one mistake on a sentence re-selected from the previous round, they are not asked to do further annotations. The annotators are encouraged to ask clarifying questions during the training rounds as well as the actual

[10] https://www.upwork.com/

annotations. If they are uncertain about the correct label for any example, they are asked to indicate this in their comments. Finally, one of the authors goes over a random selection of examples to ensure quality and also over the ones marked as uncertain to correct if necessary.

### 4.2.3 Data Splits

We split sentences in each news topic into training, development and test splits in the ratio 35:15:50. The proportions are different from the typical 80:10:10 splits but ensure that there are a sufficient number of test examples in each topic for stable and reliable results. The number of sentences and entities in each topic are shown in Table 11.

## 4.3 Results

We finetune BERT-large-cased (Devlin et al., 2019) on each topic, evaluating on all others. Hyperparameter details are listed in the appendix. We report micro-F1 at the span-level averaged over three runs with different seeds. The full evaluation table is shown in the appendix for reference. Here we discuss the aggregated results. Since domain is used to refer to the genre of text (news in this case), we use the term sub-domain to refer to the news topics. However, we still use in-subdomain (InD) and out-of-subdomain (OOD) to refer to in-subdomain and out-of-subdomain training and evaluation in the following sections.

### 4.3.1 Evaluation Sub-domain Difficulty

First, we report the performance on each test sub-domain, when a model is trained on sentences from the same sub-domain and when trained on sentences from a different sub-domain. The goal is to determine if it is easier to recognize entities in some sub-domains. The results are shown in Table 12. InD refers to the models trained on the same sub-domain as the test, and OOD refers to models trained on each of the remaining sub-domains. The OOD mean and median are aggregated over the eight models trained on each of the remaining sub-domains. As expected, in-subdomain training results in incredibly high F1 on all sub-domains. The F1 with ODD training is lower than that for in-subdomain, especially when

| tst-dom | InD | OOD | | | | | |
|---|---|---|---|---|---|---|---|
| | | mean | median | min | | max | |
| | F1 | F1 | F1 | F1 | trn-dom | F1 | trn-dom |
| arts | 92.1 | 86.9 | 88.3 | 78.4 | classified | 89.7 | metropolitan |
| business | 95.7 | 88.2 | 90.9 | 72.0 | classified | 93.2 | metropolitan |
| classified | 94.7 | 77.7 | 76.7 | 67.1 | foreign | 90.4 | editorial |
| editorial | 96.4 | 88.7 | 93.0 | 67.0 | classified | 94.6 | national |
| foreign | 96.9 | 87.5 | 92.5 | 64.2 | classified | 93.9 | national |
| metropolitan | 95.0 | 89.2 | 90.8 | 78.4 | classified | 92.8 | national |
| national | 96.2 | 90.9 | 93.8 | 79.8 | classified | 94.9 | metropolitan |
| sports | 94.8 | 81.0 | 81.0 | 77.9 | national | 84.6 | arts |
| others | 92.0 | 87.4 | 89.0 | 76.7 | classified | 90.8 | metropolitan |

Table 12: F1 on each test sub-domain, one per row, with models trained on different domains. Each row represents a test sub-domain. InD is the F1 with in-subdomain training. OOD mean and median are over the remaining eight training domains. Min and max show the F1 and training sub-domain with minimum and maximum F1 on the given test sub-domain.

testing on classified and sports. For OOD, we also report the minimum and maximum F1 on each test sub-domains, along with the corresponding training sub-domain, showing that the range of F1 also varies considerably. The lowest test F1 on most sub-domains occurs with the model trained on classified, and the highest occurs with training on national or metropolitan. For a better understanding of the variation in the performance on a given test sub-domain with different OOD sub-domains, we also show box plots (Figure 4) for the test sub-domains of classifieds and national. Depending upon the sample of sub-domains in the test set, the model performance can vary significantly even in such a homogeneous domain, leading to an incorrect characterization of the domain/dataset difficulty.

### 4.3.2 Training Sub-domain Quality

Next, we report the performance of models trained on each sub-domain when tested on the same sub-domain and on other sub-domains. The goal is to determine if it is better (or worse) to train on certain sub-domains for good performance overall. The results are shown in Table 13. InD refers to testing a model on the same sub-domain as the training data, and OOD refers to testing it on the remaining eight sub-domains. The OOD mean and median are aggregated over the eight OOD sub-domains. As expected, in-subdomain testing results

| trn-dom | InD | OOD | | | | | |
|---|---|---|---|---|---|---|---|
| | | mean | median | min | | max | |
| | F1 | F1 | F1 | F1 | tst-dom | F1 | tst-dom |
| arts | 92.1 | 87.7 | 90.4 | 73.2 | classified | 91.6 | business |
| business | 95.7 | 88.7 | 90.0 | 78.1 | classified | 94.0 | editorial |
| classified | 94.7 | 74.3 | 77.5 | 64.2 | foreign | 79.8 | national |
| editorial | 96.4 | 89.4 | 90.3 | 80.4 | sports | 94.2 | national |
| foreign | 96.9 | 85.8 | 88.8 | 67.1 | classified | 93.6 | national |
| metropolitan | 95.0 | 90.6 | 92.0 | 84.0 | sports | 94.9 | national |
| national | 96.2 | 89.2 | 91.1 | 77.9 | sports | 94.6 | editorial |
| sports | 94.8 | 82.4 | 85.1 | 68.8 | classified | 86.6 | national |
| others | 92.0 | 89.2 | 92.3 | 75.3 | classified | 94.1 | editorial |

Table 13: F1 of each training sub-domain, one per row, across different test sub-domains. Each row represents a training sub-domain. InD is the F1 for in-subdomain testing. OOD mean and median are over the remaining eight test domains. Min and max show the F1 and test sub-domain with minimum and maximum F1 for the given training sub-domain.

in incredibly high F1 on all sub-domains. The F1 with ODD testing is lower than that for in-subdomain, especially for models trained on classified and sports. For OOD, we also report the minimum and maximum F1 obtained by each model along with the corresponding test sub-domain, showing that the range of F1 also varies significantly. The lowest F1 for most models occurs when tested on classified or sports, and the highest F1 occurs when tested on national or editorial. For a better understanding of the variation in the performance of a model trained on sub-domains when tested on other sub-domains, we also show box plots (Figure 4) for the training sub-domains of classified and national. Depending upon the sample of sub-domains in the training set, the model performance can vary significantly even in such a homogeneous domain, leading to a much better or worse resulting model.

Classified and Sports stand out, exhibiting lower performance than other sub-domains for both training and testing. Examples sentences for both are shown in Table 14. Classified has several sentences that have atypical sentence structures, beginning with the last name in uppercase. For Sports, the entity type cannot be determined from the sentence-level context in several cases. In the example, it is hard to say whether the entities are names of person, location or team (organization). If this ambiguity of these entities isn't captured in

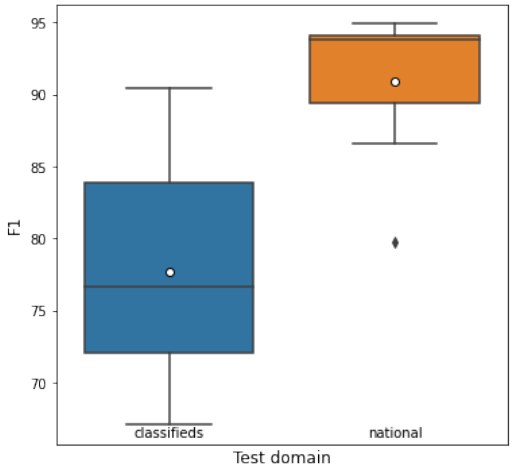



Figure 4: Box plot for two test sub-domains (classifieds and national) showing the range of F1 with training on OOD sub-domains

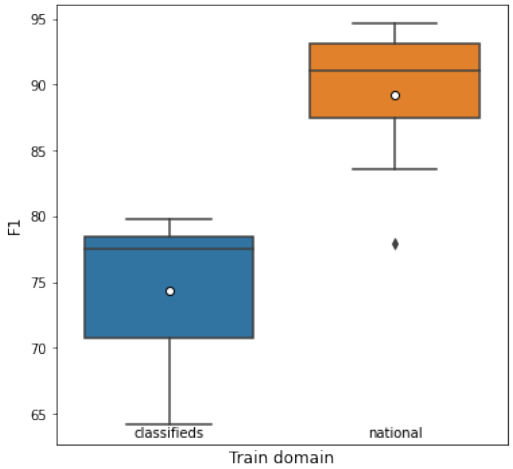



Figure 5: Box plot for two training sub-domains (classified and national), showing the range of F1 when tested on these as OOD sub-domains.

the training data, labeling them correctly is unlikely.

## 4.4 Data Selection

Datasets are typically collected by selecting some documents and then annotating all sentences in each document. The training set in CoNLL'03 (Tjong Kim Sang and De Meulder, 2003) has 15k sentences from 946 documents, Wikigold (Balasuriya et al., 2009) has 1.7k sentences from 145 pages, and MUC-7 (Chinchor, 1998) has 3.5k sentence from 100 articles.[11] This method of data selection is reasonable and intuitive. It also supports the development of models that utilize document-level context (Ratinov and Roth, 2009) which can help resolve the entity types in sentences such as the above example from sports. However, most commonly used models are built at the sentence level and the selection of full documents could result in performance similar to a model trained on the same sub-domain, with all sentences in a document representing the same sub-domain and fewer chances to cover rare sub-domains (types of documents). To illustrate this, we train models for NER

[11] MUC-7 consists of sentences from the New York Times. However, we were unable to map the documents in MUC-7 to the NYT Annotated Corpus. Regardless, MUC-7 consists only of articles on aircraft accidents and launch events, and would likely not span enough sub-domains for our analysis.

| Domain | Sentence |
|---|---|
| Classified | WEISER–Joel, passed away on March 31st, 2007. |
| Sports | Pollin clashed with Jordan at a bargaining session during the long labor standoff in November 1998. |

Table 14: Example of sentences by sub-domain

using CoNLL '03. We randomly select 3,000 training sentences as this is roughly the number of sentences in each of the sub-domains. We train three models with different seeds and report the average F1 in the third column of Table 15. CoNLL consists of news on mainly business, national, foreign and sports. Therefore, F1 on these sub-domains is closer to that with in-subdomain training, and F1 on the remaining sub-domains is close to that with out-of-subdomain training.

It is therefore essential to ensure a diverse set of sentences in the training data. Even a small number of sentences of each sub-domain in the training data can make a vast difference. Columns 'C' and 'N' in Table 15 show the F1 on various test sub-domains with a model trained on just classified or just national news. In columns 'C+10' and 'N+10', we add just 10 sentences from each of the remaining eight sub-domains. For classified, this affects each of the test sub-domains with an improvement of up to 12 points F1. On national, this mainly improves F1 on classified by 10 points and that on sports by 2 points. These two sub-domains, as shown above, exhibit different properties than the rest of the data and therefore including even a few relevant examples helps the models substantially.

One way to select relatively diverse sentences is by data selection at the sentence level instead of the document level. First, segment each document in a corpus into sentences and then select sentences randomly. While new future domains or those that evolve significantly will still be missed, this method would result in the selection of some representative samples of each existing domain. Such explicit sentence selection has been performed for domains such as Twitter where explicit documents[12] do not exist. Derczynski et al. (2016) selects tweets from different countries and different types of user accounts for linguistic variations

[12]A thread could be considered a document.

| | InD | OOD | CoNLL | C | C+10 | N | N+10 | Rndm |
|---|---|---|---|---|---|---|---|---|
| arts | 92.1 | 86.9 | 85.8 | 78.4 | 82.4 | 88.7 | 88.4 | **90.6** |
| business | 95.7 | 88.2 | 91.4 | 72.0 | 83.8 | 91.9 | 92.2 | **93.9** |
| classified | 94.7 | 77.7 | 64.8 | **94.7** | 94.6 | 83.6 | 90.2 | 93.9 |
| editorial | 96.4 | 88.7 | 89.2 | 67.0 | 83.7 | **94.6** | 94.4 | 93.7 |
| foreign | 96.9 | 87.5 | 90.4 | 64.2 | 82.6 | 93.9 | **94.0** | 93.2 |
| metropolitan | 95.0 | 89.2 | 89.0 | 78.4 | 83.5 | **92.8** | **92.8** | 91.8 |
| national | 96.2 | 90.9 | 90.0 | 79.8 | 86.2 | 96.2 | **96.3** | 93.0 |
| sports | 94.8 | 81.0 | 89.7 | 78.3 | 80.1 | 77.9 | 79.7 | **91.7** |
| others | 92.0 | 87.4 | 86.3 | 76.7 | 82.2 | **90.3** | 90.1 | 90.2 |
| Avg | 94.9 | 86.4 | 86.3 | 76.6 | 84.4 | 90.0 | 90.9 | **92.4** |

Table 15: F1 on each test sub-domain with different models. InD is in-domain training and OOD is the average of out-of-domain training. CoNLL refers to training on CoNLL '03. C and N are trained on classified and national only. C+10 and N+10 additionally include 10 sentences from each sub-domain. Rndm is random selection of sentences from a corpus with sentences in the same proportion of sub-domains as the full NYT corpus. Highest F1 in each row (excluding InD) is boldfaced.

and topics. They also account for temporal variation taking tweets from different years, months, weeks and days.

We build a model with this random sentence selection scheme. We first downsample the combined training data such that it follows the same distribution of sub-domains as the full NYT annotated corpus with 20 years of articles. This results in 10,500 training sentences and 4,494 development sentences with 14% arts, 11% business, 3% classified, 5% editorial, 7% foreign, 11% metropolitan, 8% national, 11% sports and 30% others. We then select 3,000 training and 1,284 development sentences randomly from this set. This is roughly the average number of sentences in each of the sub-domains and seeks to eliminate the impact of the training data size. Every sub-domain has at least 39 sentences in the selected training set. By training the model on this dataset, the average performance is almost the same as in-subdomain training (column rndm in Table 15).

## 4.5 Limitations

We develop a new corpus for a standard NER task, drawn from a reputable news source, New York times. Our analysis is based on the sub-domains available in the metadata of the news article. To extend it to other datasets, automatic predictors of domain are necessary. Furthermore, for a random sentence selection that includes all representative samples, a corpus spanning the entire space of sentences is needed. This is straightforward for newspapers or Wikipedia, but infeasible for domains such as Reddit or Twitter. In such cases, domain knowledge is used to select diverse sentences (Derczynski et al., 2016), again pointing to the need for automatic domain prediction. We performed domain classification experiments on our dataset via unsupervised clustering as well as zero-shot classification[13] (Yin et al., 2019), using both the known domains from the metadata and dummy domains as candidates. The accuracy of the best classifier on our data was only 30%, insufficient for better performance than a random sentence selection.

## 4.6 Conclusion

We collected a dataset for named entity recognition in a domain that is generally treated as a single domain. With models built and evaluated on sub-domains within this dataset, we show that there can be significant performance differences even within a seemingly homogenous domain. It is important to perform fine-grained inspection and careful training data selection, even within such a domain.

[13] https://huggingface.co/facebook/bart-large-mnli

# CHAPTER 5

# Temporal Effects

*This chapter is based on content originally published in: Oshin Agarwal and Ani Nenkova, 2022, Temporal effects on pre-trained models for language processing tasks. Transactions of the Association for Computational Linguistics, 10:904–921. (Agarwal and Nenkova, 2022). We evaluate the robustness of models for Named Entity Recognition (and other tasks) to changes over time. We establish terminology, as well as training and evaluation methodology to study the effect of temporal changes on pre-trained models. Most importantly, we differentiate between two aspects of temporal effects–model deterioration and the possibility of model adaptation.*

## 5.1 Introduction

Language models capture properties of language, such as the semantics of words and phrases and their typical usage, as well as facts about the world expressed in the language sample on which they were trained. Effective solutions for many language tasks depend, to a varying degree, on the background knowledge encoded in language models. Performance may degrade as language and world-related facts change. In some scenarios, language will change as a result of deploying a system that uses the language to make a prediction, as in spam detection (Fawcett, 2003). But most change is not driven by such adversarial adaptations: the language expressing sentiment in product reviews (Lukes and Søgaard, 2018), the named entities and the contexts in which they are discussed on social media (Fromreide et al., 2014; Rijhwani and Preotiuc-Pietro, 2020) and language markers of political ideology (Huang and Paul, 2018) all change over time.

Whether and how this change impacts the performance of different language technologies is a question of great practical interest. Yet research on quantifying how model performance changes with time has been sporadic. Moreover, approaches to solving language tasks have evolved rapidly, from bag-of-words models which rely on a small number of fixed words represented as strings, without underlying meaning, to fixed dense word representations such as word2vec and GloVe (Mikolov et al., 2013; Pennington et al., 2014a) and large contextualized representations of language (Peters et al., 2018; Devlin et al., 2019) that are trained on task-independent text to provide a backbone representation for word meaning. The swift change in approaches has made it hard to understand how representations and the data used to train them modulate the changes in system performance over time.

We present experiments (§5.4 & §5.5) designed to study temporal effects on downstream language tasks, disentangling worsening model performance due to temporal changes (*temporal model deterioration*) and the benefit from retraining systems on temporally more recent data in order to obtain optimal performance (*temporal domain adaptation*). We present experiments on four tasks for English. For named entity recognition, we work with an existing dataset of temporally-stratified tweets annotated with entities. Since collecting another NER dataset for robust experiments can be time-consuming, and temporal changes can affect other language tasks too, we also work with three other tasks, namely sentiment classification, truecasing and domain classification, that do not require manually labeled data and can be mined from existing data sources. We work only with tasks where the correctness of the label is not influenced by time, unlike other tasks such as open domain question answering where the answer may depend on the time when the question is posed (e.g. who is the CEO of X?). For each task, we analyze how the performance of approaches built on pre-trained representations changes over time and how retraining on more recent data influences it (§5.6). We find that models built on pre-trained representations do not experience temporal deterioration on all tasks. However, temporal domain adaptation is still possible, i.e. performance can be further improved by retraining on more recent human-labeled data.

We further find that neural models fine-tuned on the same data but initialized with random vectors for word representation exhibit dramatic temporal deterioration on the same datasets (§5.7). Models powered by pre-trained language models however are not impacted in the same way. Unlike in any prior work, we study several representations (§5.8) including those built using the same architecture and data but different model sizes (§5.10).

Even though the pre-training data of several representations overlaps in time with task-specific data and some confounding is possible, two sets of experiments show that it is unlikely (§5.9). These results provide strong evidence for model deterioration without pre-training; it also raises questions for future work, on how the (mis)match between task and pre-training data influences performance, with greater mismatch likely to be more similar to random initialization, resulting in a system more vulnerable to temporal deterioration.

The central insight from our work is that the performance of pre-trained models on downstream tasks where answer correctness is time-independent, does not necessarily deteriorate over time but that the best performance at a given time can be obtained by retraining the system on more recent data. Furthermore, based on the experiments to assess the impact of different components of a model, we provide recommendations for the design of future studies on temporal effects (§5.12). This will make it both easier to conduct future studies and have more robust findings by controlling confounding factors and ignoring others.

Finally, we present two methods for temporal adaptation that do not require manual labeling over time (§5.11). One of the methods is based on continual pre-training where we modify the typical domain adaptative pre-training with an additional step. The second method uses self-labeling and is highly effective with consistent improvement across all settings. On one of the datasets, self-labeling is even superior to fine-tuning on new human-annotated data.

## 5.2 Background

Language changes over time (Weinreich et al., 1968; Eisenstein, 2019; McCulloch, 2020). For longer time periods, a robust body of computational work has proposed methods for modeling the changes in active vocabulary (Dury and Drouin, 2011; Danescu-Niculescu-Mizil et al., 2013) and in the meaning of words (Wijaya and Yeniterzi, 2011; Hamilton et al., 2016; Rosenfeld and Erk, 2018; Brandl and Lassner, 2019). Changes in vocabulary and syntax, approximated by bi-grams in Eisenstein (2013), also occur on smaller time scales, such as days and weeks, and occur more in certain domains, e.g. change is faster in social media than in printed news. Such language changes over time can also be approximated by the change in language model perplexity. Lazaridou et al. (2021) find that language model perplexity changes faster for politics and sports than for other domains, suggesting that these domains evolve faster than others. They also demonstrate that language models do not represent well language drawn from sources published after it was trained: perplexity for text samples drawn from increasingly temporally distant sources increases steadily. Their qualitative analysis shows that the changes are not only a matter of new vocabulary: even the context in which words are used changes.

The global changes captured with language model perplexity and analysis of individual words cannot indicate how these changes impact the performance of a model for a given task. Röttger and Pierrehumbert (2021) presents a meticulously executed study of how domain change (topic of discussion) influences both language models and a downstream classification task. They show that even big changes in language model perplexity may lead to small changes in downstream task performance. They also show that domain adaptation and temporal adaptation are both helpful for the downstream classification task they study, with domain adaptation providing the larger benefit.

Here, we also focus on the question of how time impacts downstream tasks. Studying temporal change in model performance requires extra care in experimental design to tease

apart the temporal aspect from all other changes that may occur between two samples of testing data. Teasing apart temporal change from domain change is hardly possible. Even data drawn from the same source may include different domains over time[14]. Despite these difficulties, there are two clear and independent questions that we pursue, related to system performance over time.

### 5.2.1 Does performance deteriorate over time?

To study this question of *temporal model deterioration*, we need to measure performance over several time periods. Let $d_s$, $d_t$ and $d_n$ denote respectively the first, $t^{th}$ ($s \leq t \leq n$) and last temporal split in a dataset. To guard against spurious conclusions that reflect specifics of data collected in a time period, the starting point for the analysis should also vary. Huang and Paul (2018) use such a setup, performing an extensive evaluation by training $n$ models on $d_s$ to $d_n$ and then evaluating them on all remaining $n-1$ splits, both on past and future time periods. The resulting tables are cumbersome to analyze but give a realistic impression of the trends. We adopt a similar setup for our work, reporting results for a number of tasks with models trained on data from different time periods and tested on data from all subsequent time periods available. In addition, we introduce summary statistics that capture changes across all studied time periods to compare the temporal trends easily and to compute statistical significance for the observed changes in performance (§5.4 & 5.5).

Most prior work, in contrast, uses a reduced setup (Lukes and Søgaard, 2018; Rijhwani and Preotiuc-Pietro, 2020; Søgaard et al., 2021) with a fixed test time period and measures the performance of models trained on different time periods on this fixed test set. Such evaluation on one future temporal split does not measure the change in model performance over time and cannot support any conclusions about temporal deterioration.[15] This setup from prior work supports conclusions only about temporal domain adaptation i.e. whether

[14] Huang and Paul (2018) find the topics in their data and observe that the top 20 topics change over time.

[15] Lazaridou et al. (2021) omit such an evaluation since they measure language model perplexity which is sensitive to document length, which they found differed across months. Röttger and Pierrehumbert (2021) evaluate over multiple test sets on a classification task but also omit such an evaluation by reporting the change in the metrics of models w.r.t. a control model without temporal adaptation.

retraining on temporally new data helps improve performance on future data, with a single-point estimate for the improvement.

### 5.2.2 Can performance at time $t$ be improved?

As described above, most prior work chose $d_n$, the data from latest time period as the test data, to evaluate models trained on earlier data. Lukes and Søgaard (2018) train a model for sentiment analysis of product reviews in 2001-2004 and 2008-2011 and test them on reviews from 2012-2015. Rijhwani and Preotiuc-Pietro (2020) train models for named entity recognition on tweets from each year from 2014 to 2018 and test them on tweets from 2019. Søgaard et al. (2021) work with the tasks of headline generation and emoji prediction. For headline generation, they successively train models on data from 1993 to 2003 and test it on data from 2004. For emoji prediction, the training data comes from different days and the last one is used as the test set. Lazaridou et al. (2021) train a language model on various corpora with test data from 2018-2019 and train years that either overlap with the test year or precede them.

Such results allow us to draw conclusions about the potential for *temporal domain adaptation*, revealing that models trained on data closer to the test year perform better on that test year. The only problem is that there is a single test year chosen and any anomaly in that test year may lead to misleading results. The temporal Twitter corpus (Rijhwani and Preotiuc-Pietro, 2020), where 2019 is the dedicated test year, is an instructive case in point. Twitter increased the character limit in late 2017. As a result, tweets from 2018 are longer and contain more entities than those in prior years. The potential for temporal adaptation measured only on 2018 data contrasted with prior years may give a highly optimistic view of how much models can improve. An evaluation setup like the one in Huang and Paul (2018) or the recent work in Röttger and Pierrehumbert (2021) is needed to draw robust conclusions. We adopt their setup with some changes. We also introduce summary statistics to easily interpret trends and a test for significance to determine if the changes in performance are compatible with random fluctuation of performance across time periods.

Another line of work on temporal effects focuses on temporal adaptation by incorporating time in the training process as opposed to retraining models on new human-labeled data regularly. Several approaches have been proposed such as diachronic word embeddings, the "frustratingly simple" domain adaptation, aligning representations of old and new data, time-aware self-attention and continual learning as new data is available (He et al., 2018; Huang and Paul, 2019; Bjerva et al., 2020; Hofmann et al., 2021; Rosin and Radinsky, 2022). An expanded evaluation of these approaches to measure deterioration and adaptation across several time periods with different representations will be useful, given our findings.

## 5.3 Experimental Resources

Here we describe the datasets and the different models used.

### 5.3.1 Tasks and Datasets

We use four English datasets, two for sequence labeling and two for text classification[16].

**Named Entity Recognition with Temporal Twitter Corpus** TTC (Rijhwani and Preotiuc-Pietro, 2020) consists of tweets annotated with *PER*, *LOC* and *ORG* entities. There are 2,000 tweets in each year from the period 2014–2019. TTC is the only corpus with human annotations specifically collected in order to study temporal effects on performance. Other datasets, including the three we describe next, are in fact derived annotations that do not require manual annotation.

**Truecasing with New York Times** Truecasing (Gale et al., 1995; Lita et al., 2003) is the task of case restoration in text. We sample a dataset from the NYT Annotated Corpus (Sandhaus, 2008) which has sentences that follow English orthographic conventions. We perform a constrained random sampling of 10,000 sentences per year from 1987–2004 and organize the data with three consecutive years per split. To maintain diversity of text,

[16]More dataset details and model hyperparameters can be found in the appendix and at https://github.com/oagarwal/temporal-effects

we select an approximately equal number of sentences from each domain (indicated by the metadata) and only two sentences per article. Sentences should have at least one capitalized word, not including the first word and should not be completely in uppercase (headlines appear in all uppercase). We model the task as sequence labeling with binary word labels of fully lowercase or not.

**Sentiment Classification with Amazon Reviews** AR (Ni et al., 2019) consists of 233M product reviews rated on a scale of 1 to 5. Following prior work (Lukes and Søgaard, 2018), we model this task as binary classification, treating a rating of greater than 3 as positive and the remaining as negative. We randomly sample 40,000 reviews per year from the period 2001–2018 and organize the data with three consecutive years per split. The first 50 words of each review are used.

**Domain Classification with New York Times** We select the first 40,000 articles from each year in 1987–2004 from the NYT Annotated Corpus and organize the data with three consecutive years per split. The article domain is labeled using the article metadata. Certain domains are merged based on the name overlap, resulting in eight domains —Arts, Business, Editorial, Financial, Metropolitan, National, Sports and Others. The first 50 words (1-2 paragraphs) of each article are used.

### 5.3.2 Models

We use two architectures (biLSTM-CRF and Transformers) and four representations (GloVe, ELMo, BERT, RoBERTa) for the experiments. Hyperparameters and other fine-tuning details are noted in the appendix.

**GloVe+char** BiLSTM (Hochreiter and Schmidhuber, 1997) with 840B-300d-cased GloVe (Pennington et al., 2014a) and character-based word representation (Ma and Hovy, 2016) as input. For sequence labeling, a CRF (Lafferty et al., 2001) layer is added and prediction is made for each word. For text classification, the representation of the first word is used to

make the prediction.

**ELMo+GloVe+char** Same as GloVe+char but the Original ELMo (Peters et al., 2018) embeddings are concatenated to the input.

**BERT** (Devlin et al., 2019) We use the large model for sequence labeling and the base model for text classification, both cased. The number of training examples was larger for text classification resulting in a much faster base model with minimally lower performance than the large one.

**RoBERTa** (Liu et al., 2019b) We use the large model for sequence labeling and the base model for text classification.

## 5.4 Experimental Setup

We divide each dataset into $n$ temporal splits with an equal number of sentences for sequence labeling and an equal number of documents for text classification to minimize any performance difference due to the size of the split. We randomly downsample to the size of the smallest temporal split whenever necessary. Let $d_s$, $d_t$ and $d_n$ denote the first, $t^{th}$ and last temporal split in the dataset respectively.

**Train and Test Set** We largely follow Huang and Paul (2018), with minor clarifications on certain aspects as well as additional constraints due to differences in dataset size across tasks, ensuring consistency in setup. First, we vary both training and test years but limit the evaluation to future years since we want to mimic the practical setup of model deployment. We train $n-1$ models, each on a specific temporal split, starting from a model on $d_s$ to a model on $d_{n-1}$, and evaluate the model trained on $d_t$ on test sets starting from $d_{t+1}$ to $d_n$. Each temporal split has the same number of sentences/documents and training/evaluation is done only on data from a given split (not cumulative data). An increase in training data size is typically associated with an increase in performance, so cumulative expansion of the

training set will introduce a confound between the temporal effects and dataset size. With these results, a lower triangular matrix can be created with the training years as the columns and the test years as the rows. A sample can be seen in Table 16.

Next, we need to further divide each temporal split $d_t$ into three sub-splits for training, development and testing. We are limited by our smallest dataset on NER, which is by far the hardest to label and is the only task that requires manual data annotation. It has 2,000 sentences in each year and splitting it into three parts will not provide us with enough data to train a large neural model or reliably evaluate it. Hence, we do not evaluate on the current year but only on the future ones. When training a model on $d_t$, it is split 80-20 into a training set $train_t$ and a development set $dev_t$. Both these sets combined i.e. the full $d_t$ serves as the test set $test_t$ when a past model is evaluated on it.

**Development Set** The model checkpoint that performs best on the development set is typically chosen as the model to be tested. Yet prior work either does not report full details of the data used for choosing hyperparameters (Lukes and Søgaard, 2018), or uses default hyperparameters (Fromreide et al., 2014), or draws the development set from the same year as the test set year (Rijhwani and Preotiuc-Pietro, 2020; Chen et al., 2021; Søgaard et al., 2021). We choose development data from the time period of the training data, reserving 20% of the data in each temporal split since data from a future time period will not be available to use as the development set during training. Beyond concerns about setup feasibility, through experiments, we found that the selection of development set from the test year may affect performance trends and even lead to exaggerated improvement for temporal domain adaptation.

## 5.5 Evaluation Metrics

**Task Metrics** In the full matrix described above, we report task-specific metrics, by averaging them over three runs with different random seeds. For NER, we report the span-level micro-F1 over all entity classes; for truecasing, we report F1 for the cased class. For senti-

ment classification, we report F1 for the negative sentiment class; for domain classification, we report the macro-F1 over all the domains. The positive sentiment and uncased word account for about 80% of the data in their respective tasks and are largely (but not completely) unaffected over time.

**Temporal Summary Metrics** For a compact representation, we also report summary deterioration score (DS) and summary adaptation score (AS) in addition to the full matrix with the task-specific evaluation results. Deterioration score measures the average change in the performance of a model over time. A negative score indicates that the performance has deteriorated. Similarly, the adaptation score measures the average improvement in performance by retraining on recent data, labeled or unlabeled (§5.11). A positive score means performance improves by retraining.

For each score, we report two versions, one that measures the average change between immediately consecutive time periods and the other that measures the change with respect to an anchor (oldest) time period since retraining need not be at regular intervals. The anchor-based scores are also a more stable metric since the amount of time passed between the values being compared is longer and therefore we are more likely to observe discernible temporal effects. For measuring deterioration, the anchor is the oldest test time period for the given model i.e. if a model is trained on $d_t$, then the task metric on $d_{t+1}$ is the anchor score (first available row in each column of the full results matrix). For measuring adaptation, the anchor is the oldest train time period so the anchor is always $d_s$ (first column in the full results matrix). Let $i$ be a time period in the training set and let $j$ be a time period in the test set. Let $M_i^j$ be the task metric measured on $d_j$ when the model is trained on $d_i$. Let N be the number of elements in the sum and $d_a$ be the anchor time period. The summary scores are defined as follows.

$$DS_t^{t-1} = \frac{1}{N} \sum_{i \in train} \sum_{j \in test} M_i^{j+1} - M_i^j \tag{5.1}$$

$$DS_t^a = \frac{1}{N} \sum_{i \in train} \sum_{j \in test} M_i^{j+1} - M_i^a \tag{5.2}$$

$$AS_t^{t-1} = \frac{1}{N} \sum_{i \in train} \sum_{j \in test} M_{i+1}^j - M_i^j \tag{5.3}$$

$$AS_t^a = \frac{1}{N} \sum_{i \in train} \sum_{j \in test} M_{i+1}^j - M_a^j \tag{5.4}$$

To test if a given trend for deterioration or adaptation is statistically significant, we consider the vector of differences in each of the formulae above, and run a two-sided Wilcoxon signed-rank test to check if the median difference is significantly different from zero. For our setup there are 10 differences total, corresponding to a sample size of $N$=10. When we report deterioration and adaptation scores in tables with results, we indicate with an asterisk (*) values corresponding to a vector of differences with p-value smaller than 0.05. While this measures the fluctuations across the average task metrics over different training and test years, it does not take into account the variations across different runs of the same model with random seeds. An ideal test would take into account both the random seeds and the different train/test years. However, this is not straightforward and we leave the design of such a test for future work. Instead, in this paper, to ensure trends are not affected by variations across seeds, we calculate three values for each of the four scores, corresponding to the three runs. For deterioration, the performance of a model trained with a specific seed is measured over time, but for adaptation, the performance change may be measured w.r.t. a model trained with a different seed, as will be the case in practice. We then report the minimum and maximum of this score for the significant summary metrics as measured above. If the sign of the minimum and maximum of each score is the same, the trend in the scores remains same across runs, even if the magnitude varies.

Along with the summary scores, we also report three salient values of the task metric

| | Train Year | | | | |
|---|---|---|---|---|---|
| Test Year | 2014 | 2015 | 2016 | 2017 | 2018 |
| GloVe+char biLSTM-CRF | | | | | |
| 2015 | 55.18 | - | - | - | - |
| 2016 | 56.22 | 57.13 | - | - | - |
| 2017 | 55.09 | 53.95 | 59.43 | - | - |
| 2018 | 51.06 | 53.12 | 57.75 | 57.82 | - |
| 2019 | 54.10 | 54.56 | 59.48 | 60.41 | 62.99 |
| RoBERTa | | | | | |
| 2015 | 67.48 | - | - | - | - |
| 2016 | 69.41 | 72.02 | - | - | - |
| 2017 | 68.30 | 70.53 | 70.29 | - | - |
| 2018 | 67.82 | 68.33 | 69.29 | 68.60 | - |
| 2019 | 77.79 | 78.33 | 78.89 | 78.28 | 79.99 |

Table 16: F1 for NER on TTC. Training is on gold-standard data.

from the full results table (Table 16) in the summary ($M_s^{s+1}$, $M_s^n$ and $M_{n-1}^n$), necessary to compare the relative performance across datasets and representations. Remember that $M_i^j$ is the evaluation metric measured on $d_j$ when the model is trained on data split $d_i$. $M_s^{s+1}$, which is the value in the first row and first column in the full results table, represents the task metric when the model is trained on the first temporal split and evaluated on the immediate next one. It serves as the base value for comparison. $M_s^n$, which is the value in the last row and first column in the full results table, shows whether the performance of the model deteriorated from $M_s^{s+1}$ over the longest time span available in the dataset, by comparing the performance of the same model on the last temporal split. Similarly, $M_{n-1}^n$, which is the value in the last row and last column of the full results table, shows if the performance can be improved by retraining from $M_s^n$ over the longest time span available in the dataset, by retraining on the latest available temporal split.

## 5.6 Main Results

Results are shown in Table 16 and Table 17. For NER, we show the full matrix with the task metrics but for all other tasks, we only report the summary scores. Here, we only report results with the oldest (GloVe) and latest (RoBERTa) representations used in our

| | $M_s^{s+1}$ | $M_s^n$ | $M_{n-1}^n$ | $D_t^a$ | $A_t^a$ | $D_t^{t-1}$ | $A_t^{t-1}$ |
|---|---|---|---|---|---|---|---|
| NER-TTC | | | | | | | |
| GloVe | 55.2 | 54.1 | 63.0 | -1.3 | 4.1* | -0.1 | 2.1* |
| RoBERTa | 67.5 | 77.8 | 80.0 | 3.2 | 1.4* | 3.5 | 0.8 |
| Truecasing-NYT | | | | | | | |
| GloVe | 93.8 | 93.0 | 94.6 | -0.6* | 0.3 | -0.2* | 0.3 |
| RoBERTa | 97.5 | 94.4 | 95.6 | -1.1 | 0.4* | -0.8 | 0.2* |
| Sentiment-AR | | | | | | | |
| GloVe | 44.9 | 42.8 | 64.7 | 0.8 | 10.3* | 0.4 | 4.9* |
| RoBERTa | 69.9 | 73.9 | 78.9 | 2.5* | 2.5* | 1.3* | 1.1* |
| Domain-NYT | | | | | | | |
| GloVe | 73.0 | 68.4 | 78.1 | -2.7* | 7.9* | -0.5 | 3.6* |
| RoBERTa | 84.2 | 78.2 | 86.6 | -3.7* | 5.8* | -1.1* | 2.9* |

Table 17: Deterioration and Adaptation scores for models fine-tuned on gold standard data. Positive and negative scores denotes an increase and decrease in the task metric respectively. An asterisk marks statistically significant scores.

experiments. For other representations, we provide a detailed analysis in later sections.

**Temporal Model Deterioration** can be tracked over the columns in the full matrix and by comparing $M_s^{s+1}$ and $M_s^n$ along with the deterioration scores in the summary table. Each column in the full matrix presents the performance of a fixed model over time on future data. We do not observe temporal deterioration in all cases. For NER, we observe deterioration with GloVe but not with RoBERTa, for which performance improves over time. However, neither of the deterioration scores are statistically significant. For sentiment, there is no deterioration; in fact model performance improves over time (significant for RoBERTa). The improvement in performance over time can be attributed to the change in task difficulty. The character limit for tweets was doubled in 2017, allowing for more context and therefore easier NER. Product reviews on Amazon became increasingly shorter over time making it easier to recognize the sentiment expressed in them. For truecasing, there is some deterioration (significant for GloVe). For domain classification, there is considerable deterioration (significant for both representations). The difference between the two versions of the deterioration scores is as expected, smaller for consecutive periods and larger when computed with respect to the anchor.

| | $D_t^a$ | $A_t^a$ | $D_t^{t-1}$ | $A_t^{t-1}$ |
|---|---|---|---|---|
| | | NER-TTC | | |
| GloVe | - | [2.5, 7.0] | - | [1.4, 3.3] |
| RoBERTa | - | [0.7, 1.9] | - | - |
| | | Truecasing-NYT | | |
| GloVe | [-0.6, -0.5] | - | [-0.2, -0.2] | - |
| RoBERTa | - | [0.3, 0.4] | - | [0.2, 0.2] |
| | | Sentiment-AR | | |
| GloVe | - | [6.9, 14.0] | - | [3.7, 6.1] |
| RoBERTa | [2.4, 2.5] | [1.8, 3.0] | [1.2, 1.4] | [0.8, 1.3] |
| | | Domain-NYT | | |
| GloVe | [-3.1, -2.1] | [6.8, 8.7] | - | [3.3, 4.0] |
| RoBERTa | [-3.9, -3.6] | [5.4, 6.0], | [-1.2, -1.1] | [2.8, 3.0] |

Table 18: Minimum and maximum of summary scores across three runs for models fine-tuned on gold-standard data, for statistically significant summary metrics. Both have the same sign, showing the trends remains the same across runs.

Model deterioration appears to be both task and representation dependent. This result offers a contrast to the findings in Lazaridou et al. (2021) that language models get increasingly worse at predicting future utterances. We find that not all tasks suffer from model deterioration. The temporal change in vocabulary and facts does not affect all tasks as these changes and information might not be necessary to solve all tasks. These results do not depend on whether pre-training data and task data overlap temporally (§**??**).

**Temporal Domain Adaptation** can be tracked over the rows in the full matrix and by comparing $M_s^n$ and $M_{n-1}^n$ along with the adaptation scores in the summary table. Each row in the full matrix represents performance on a fixed test set starting with models trained on data farthest away to the temporally nearest data. Performance improves with statistical significance as the models are retrained on data that is temporally closer to the test year. The results are consistent with prior work that uses non-neural models (Fromreide et al., 2014; Lukes and Søgaard, 2018; Huang and Paul, 2018) or evaluates on a single test set (Lukes and Søgaard, 2018; Rijhwani and Preotiuc-Pietro, 2020; Søgaard et al., 2021). However, the extent of improvement varies considerably by test year, task and representation. The largest improvement is for the domain classification followed by the sentiment classification. It is

| | $M_s^{s+1}$ | $M_s^n$ | $M_{n-1}^n$ | $D_t^a$ | $A_t^a$ | $D_t^{t-1}$ | $A_t^{t-1}$ |
|---|---|---|---|---|---|---|---|
| NER | 21.8 | 10.1 | 23.9 | -6.6* | 6.4* | -2.7* | 3.4* |
| Truecasin | 89.0 | 86.0 | 88.2 | -1.5* | 0.7 | -0.7* | 0.5 |
| Sentiment | 41.6 | 37.7 | 59.7 | -0.2 | 9.0* | -0.3 | 4.2* |
| Domain | 59.7 | 48.0 | 68.6 | -5.7* | 16.7* | -2.1* | 7.2* |

Table 19: Deterioration and Adaptation scores for biLSTM with randomly initialized word representations fine-tuned on gold standard data. An asterisk marks statistically significant scores.

worth noting that both of these datasets span 18 years whereas the NER dataset spans 6 years and more improvement may be observed for NER for a similar larger time gap. The change in performance on truecasing is almost non-existent. The difference between the two versions of the adaptation scores is as expected given the longer gap between retraining.

For all four summary scores, we also report the minimum and maximum by calculating three values of each score corresponding to three different runs (§5.5). The results are shown in Table 18. While the extent of deterioration and adaptation varies across runs, the sign of the scores is the same for the maximum and minimum score, i.e. the trends are consistent across runs.

## 5.7 No Pre-training

Above we found that models powered by pre-trained representations do not necessarily manifest temporal deterioration. At first glance, our findings may appear to contradict findings from prior work. They appear more compatible though when we note that most of the early work discussing temporal effects on model performance studied bag of words models (Fromreide et al., 2014; Lukes and Søgaard, 2018; Huang and Paul, 2018). Given that bag-of-word models are rarely used now, we do not perform experiments with them. Instead, we provide results with biLSTM representations initialized with random vectors for word representations. These learn only from the training data and their performance mirrors many of the trends reported in older work. The results are shown in Table 19. Contrary to the results with pre-trained representations, most deterioration scores are negative, large

in magnitude and statistically significant. The change in task difficulty which increased model performance plays a role only when there is a strong background representation of the text. Adaptation scores are consistent i.e. positive and statistically significant but have larger magnitudes than those with pre-trained representations. Pre-training on unlabeled data injects background knowledge into models beyond the training data and has led to significant improvement on many NLP tasks. It also helps avoid or reduce the extent of temporal deterioration in models, making deployed models more (though not completely) robust to changes over time.

## 5.8 Different Pre-trained Representations

Given the variety of language representations, it is tempting to choose one for experimentation and assume that findings carry over to all. We present results using four different representations and find that this convenient assumption does not bear out in practice. We use popular representations that are likely to be used out-of-the-box.[17]

Both temporal model deterioration and temporal domain adaptation vary vastly across representations (Table 20). RoBERTa stands out as the representation for which results deteriorate least and for which the potential for temporal adaptation is also small. RoBERTa exhibits significant deterioration with respect to the anchor only for domain prediction; it significantly improves over time for sentiment prediction, and changes are not significant for NER and truecasing. The improvements from temporal adaptation with respect to the anchor are statistically significant for all tasks for RoBERTa, but smaller in size compared to the improvements possible for the other representations. GloVe in contrast shows performance deterioration with respect to the anchor for three tasks (NER, truecasing and domain prediction), significant for the last two; on sentiment analysis performance of the GloVe model improves slightly, but not significantly while all other representations show

[17] Admittedly, input representation is an overloaded term that encompasses the model architecture, whether the representation is contextual or not, what data is used for pre-training, the overall model size, the length of the final vector representation, etc. We discuss several of these differentiating factors later.

| | $M_s^{s+1}$ | $M_s^n$ | $M_{n-1}^n$ | $D_t^a$ | $A_t^a$ | $D_t^{t-1}$ | $A_t^{t-1}$ |
|---|---|---|---|---|---|---|---|
| NER-TTC | | | | | | | |
| GloVe | 55.2 | 54.1 | 63.0 | -1.3 | 4.1* | -0.1 | 2.1* |
| Gl+ELMo | 59.6 | 63.1 | 68.7 | 0.7 | 1.5* | 1.0 | 1.0 |
| BERT | 64.7 | 71.7 | 76.2 | 2.7 | 1.1* | 2.9 | 0.7* |
| RoBERTa | 67.5 | 77.8 | 80.0 | 3.2 | 1.4* | 3.5 | 0.8 |
| Truecasing-NYT | | | | | | | |
| GloVe | 93.8 | 93.0 | 94.6 | -0.6* | 0.3 | -0.2* | 0.3 |
| Gl+ELMo | 94.4 | 93.4 | 95.1 | -0.6* | 0.5* | -0.3* | 0.3* |
| BERT | 97.2 | 94.0 | 94.6 | -1.1 | 0.3* | -0.8 | 0.2* |
| RoBERTa | 97.5 | 94.4 | 95.6 | -1.1 | 0.4* | -0.8 | 0.2* |
| Sentiment-AR | | | | | | | |
| GloVe | 44.9 | 42.8 | 64.7 | 0.8 | 10.3* | 0.4 | 4.9* |
| Gl+ELMo | 55.3 | 57.5 | 69.1 | 2.6* | 5.5* | 1.2* | 2.2* |
| BERT | 63.1 | 65.9 | 75.2 | 2.4* | 4.7* | 1.3* | 2.0* |
| RoBERTa | 69.9 | 73.9 | 78.9 | 2.5* | 2.5* | 1.3* | 1.1* |
| Domain-NYT | | | | | | | |
| GloVe | 73.0 | 68.4 | 78.1 | -2.7* | 7.9* | -0.5 | 3.6* |
| Gl+ELMo | 77.9 | 70.7 | 82.8 | -3.9* | 9.4* | -1.0 | 4.3* |
| BERT | 82.7 | 74.3 | 86.2 | -4.6* | 9.4* | -1.3* | 4.2* |
| RoBERTa | 84.2 | 78.2 | 86.6 | -3.7* | 5.8* | -1.1* | 2.9* |

Table 20: Deterioration and Adaptation scores for models fine-tuned on gold standard data with various input representations. An asterisk marks statistically significant scores.

significant improvements over time.

The tables also allow us to assess the impact of using a new (more recent state-of-the-art) pre-trained representation vs. annotating new training data for the same pre-trained representation. An approximate comparison can be made between two representations A and B where A is the older representation, by comparing $M_n^{n-1}$ of A i.e. training on $n-1$ instead of $s$ to $M_s^n$ of B i.e. still training on $s$ but using representation B. For example, consider NER with $M_s^n$ using GloVe as 54.1. By retraining the model with new training data, the F1 obtained is 63.0. However, by using newer representation of GloVe+ELMO, BERT and RoBERTa with the old training data, the F1 us 63.1, 71.7 and 77.8 respectively. The benefit of using new training data vs. new pre-trained representations is again highly dependent on the task and representation.

| Model/Task | Corpus | Time Span |
|---|---|---|
| | Task Dataset | |
| NER | TTC | 2014-2019 |
| Truecasing | NYT | 1987-2004 |
| Sentiment | Amazon Reviews | 2001-2018 |
| Domain | NYT | 1987-2004 |
| | Pretraining Data | |
| GloVe | Common Crawl | till 2014* |
| ELMo | 1B Benchmark | till 2011* |
| | Wikipedia | Jan 2001-2018* |
| BERT | BookCorpus | till 2015* |
| | Wikipedia | Jan 2001-2018* |
| | BookCorpus | till 2015* |
| | CC-News | Sept 2016-Feb 2019 |
| | OpenWebText | till 2019* |
| RoBERTa | Stories | till 2018* |

Table 21: Time Span for all datasets and corpora. All corpora only include English data. * denotes that the actual time span is unknown so we note the publication date of the dataset/paper instead.

## 5.9 Pre-training Data Time Period

To perform clear experiments, one would need to control the time period of not only the task dataset but also the pre-training corpora. We report the time span for each dataset and the pre-training corpus of each model in Table 21. For several corpora, the actual time span is unknown so we report the time of dataset/paper publication instead. This table makes it easy to spot where a cleaner experimental design may be needed for future studies.

Most pre-training corpora overlap with the task dataset time span, making it hard to isolate the impact of temporal changes. BERT is trained on Wikipedia containing data from its launch in 2001 till 2018, and 11k books, spanning an unknown time period. RoBERTa uses all of the data used by BERT, and several other corpora. The pre-training data of both BERT and RoBERTa overlaps with all training and test periods of the datasets used.

We also use GloVe and ELMo representations, which do not overlap with the TTC dataset (2014-2019). GloVe was released in 2014, hence is trained on data prior to 2014.

| | All splits | Last split |
|---|---|---|
| GloVe | 0.8 | 1.0 |
| GloVe+ELMo | 2.6 | 3.1 |

Table 22: Deterioration score w.r.t. the anchor for Amazon Reviews averaged over all temporal test splits compared to the last temporal split which does not overlap with the pre-training data time period. Scores are positive for both.

ELMo uses the 1B benchmark for pre-training which has data from WMT 2011. Yet for these two, change in model performance over time is statistically insignificant, consistent with the cases when there is overlap. Adaptation scores are higher but cannot be attributed to the lack of overlap since there is a high potential for adaptation even when there is overlap for the other tasks. GloVe and ELMo also do not overlap with a portion of the Amazon Reviews dataset (2016-2018). This is the last temporal split and is therefore used only for evaluation. Since the pre-training data does not overlap with this split, model deterioration might be expected but results on this split follow the same trend of increasing F1 with time. Table 22 shows the average deterioration score w.r.t. the anchor for all splits and the 2016-2018 split, both of which are positive. Therefore, we have at least a subset of experiments free of confounds due to pre-training time overlap. The observed trends hold across both the set of experiments with and without overlap.

Additionally, for the domain classification task, the pre-training time period overlaps with the training and evaluation years, yet we still observe considerable model deterioration. Both experiments where we do not observe deterioration despite no overlap and observe deterioration despite overlap, point to the lower impact of pre-training time period on the downstream task. Instead, these results suggest that changes in performance are task-dependent and performance is most impacted by the size of the pre-training data or the differences in model architecture.

Regardless, an important set of experiments for future work would involve pre-training the best-performing model on different corpora controlled for time and comparing their performance. Such an extensive set of experiments would require significant computational

| Size | BERT | RoBERTa |
|---|---|---|
| Large | 340M | 355M |
| Base | 110M | 125M |
| Distil-base | 65M | 82M |

Table 23: Number of parameters in the models

resources as well as time. Because of this, prior work has, like ours, worked with off-the-shelf pre-trained models. For instance, Röttger and Pierrehumbert (2021) control the time period for the data used for intermediate pre-training in their experiments, but they start their experiments with BERT which is pre-trained on corpora that overlap temporally with their downstream task dataset. For future work, we emphasize the need to report the time period of any data used to support research on temporal model deterioration and temporal domain adaptation.

## 5.10 Model Size

Lastly, we assess the differences in temporal effects between models with the same architecture and pre-training data[18] but different sizes. We use three versions for BERT and RoBERTa—large, base and distil-base. The distil-base models are trained via knowledge distillation from the base model (Sanh et al., 2019). The number of parameters in each are reported in Table 23. For sequence labeling (named entity recognition and truecasing), we compare all three versions. For text classification, we do not train the large model.

Results are shown in Table 24. As expected, the smaller model sizes have lower F1 across all tasks, though the impact of the model size on the amount of deterioration and possible adaptation varies across task. In truecasing, there is no or little difference between the deterioration and adaptation scores across model sizes. For all other tasks, there is generally more deterioration (or less improvement for a positive score) and more scope for adaptation via retraining for smaller models. Nonetheless, the overall trend i.e. the direction of change in performance is consistent across model sizes.

[18] distil-RoBERTa uses less pre-training data than RoBERTa.

| | BERT | | | | | | | RoBERTa | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | $M_s^{s+1}$ | $M_s^n$ | $M_{n-1}^n$ | $D_t^a$ | $A_t^a$ | $D_t^{t-1}$ | $A_t^{t-1}$ | $M_s^{s+1}$ | $M_s^n$ | $M_{n-1}^n$ | $D_t^a$ | $A_t^a$ | $D_t^{t-1}$ | $A_t^{t-1}$ |
| | | | | | | NER-TTC | | | | | | | | |
| distil-base | 59.3 | 60.0 | 69.0 | -0.4 | 3.5* | 0.9 | 1.8* | 60.7 | 67.2 | 70.8 | 2.1 | 1.9* | 2.7 | 0.9* |
| base | 64.1 | 65.6 | 72.1 | 0.9 | 2.2* | 1.6 | 0.9* | 66.8 | 73.6 | 76.0 | 2.3 | 0.3 | 2.8 | 0.4 |
| large | 64.7 | 71.7 | 76.2 | 2.7 | 1.1* | 2.9 | 0.7* | 67.5 | 77.8 | 80.0 | 3.2 | 1.4* | 3.5 | 0.8 |
| | | | | | | Truecasing-NYT | | | | | | | | |
| distil-base | 96.9 | 93.7 | 95.1 | -1.3* | 0.4* | -0.8 | 0.3* | 96.4 | 93.0 | 94.4 | -1.2 | 0.4* | -0.8 | 0.3* |
| base | 97.1 | 93.8 | 95.2 | -1.2 | 0.4* | -0.8 | 0.3* | 97.0 | 93.8 | 95.1 | -1.2 | 0.4* | -0.8 | 0.2* |
| large | 97.2 | 94.0 | 94.6 | -1.1 | 0.3* | -0.8 | 0.2* | 97.5 | 94.4 | 95.6 | -1.1 | 0.4* | -0.8 | 0.2* |
| | | | | | | Sentiment-AR | | | | | | | | |
| distil-base | 59.9 | 62.9 | 73.7 | 2.9* | 5.7* | 1.6* | 2.4* | 65.8 | 70.1 | 76.4 | 2.8* | 3.2* | 1.4* | 1.4* |
| base | 63.1 | 65.9 | 75.2 | 2.4* | 4.7* | 1.3* | 2.0* | 69.9 | 73.9 | 78.9 | 2.5* | 2.5* | 1.3* | 1.1* |
| | | | | | | Domain-NYT | | | | | | | | |
| distil-base | 81.7 | 72.9 | 85.2 | -4.9* | 9.8* | -1.4 | 4.4* | 83.2 | 75.6 | 86.1 | -4.2* | 7.8* | -1.2 | 3.6* |
| base | 82.7 | 74.3 | 86.2 | -4.6* | 9.4* | -1.3* | 4.2* | 84.2 | 78.2 | 86.6 | -3.7* | 5.8* | -1.1* | 2.9* |

Table 24: Deterioration and Adaptation scores for different model sizes with same architecture and pre-training data, fine-tuned on gold standard data. An asterisk marks statistically significant scores.

Unlike language models which experience a similar change in perplexity over time for different model sizes (Lazaridou et al., 2021), we find that larger models show less deterioration (or more increase) and allow for less room for adaptation by retraining. Smaller models likely "memorize" less data and therefore depend more on the training data, thereby experiencing more deterioration and higher improvement via retraining. This is further substantiated by the largest change in adaptation score with model size in the task of NER where entity memorization in pre-training may play a larger role in task performance (Agarwal et al., 2021b).

## 5.11 Temporal Adaptation without New Human Annotations

We found that model deterioration and the possibility of temporal adaptation need to be distinguished and both need to be measured. Model deterioration is task-dependent where some tasks suffer from deterioration and others do not. Regardless of whether model deterioration exists or not, for all tasks, performance can be improved by retraining on human-labeled

| | NER | Truecasing | Sentiment | Domain |
|---|---|---|---|---|
| | | BERT | | |
| Gold | 1.14* | 0.29* | **4.70*** | **9.37*** |
| Pretrain | 0.84* | - | 1.43* | -0.01 |
| Self-Label | **2.27*** | - | 1.56* | 1.14* |
| | | RoBERTa | | |
| Gold | 1.39* | 0.35* | **2.49*** | **5.83*** |
| Pretrain | -0.84* | - | 0.25* | -1.34 |
| Self-Label | **1.79*** | - | 1.40* | 1.01* |

Table 25: Adaptation scores w.r.t. anchor time period for different adaptation methods. Large model is used for sequence labeling and base model for text classification.

data from a more recent time period. For tasks such as NER where the collection of new data can involve significant effort, this begets the question — how can we perform temporal adaptation without collecting new human annotations. Here, we explore methods for this. Given human annotations for $d_s$ and a model trained on it, we want to improve the performance of this model on $d_t$ without human annotations on $d_t$. For these experiments, we only use NER-TTC, Sentiment-AR and Domain-NYT since Truecasing-NYT showed little change in performance even when retrained with even gold-standard data.

### 5.11.1 Continual Pre-training

For the first experiment, we use domain adaptive pre-training (Gururangan et al., 2020) on temporal splits. A pre-trained model undergoes a second pre-training on domain-specific unlabeled data before fine-tuning on task-specific labeled data. In our case, the new unlabeled data is a future temporal split. However, unlike in typical domain adaptive pre-training, we only have a small amount of in-domain data. In practice, the amount of this data would depend upon how frequently one wants to retrain the model. For the experiments, we use the data from temporal split $d_t$, throwing away the gold-standard annotations. We take a pre-trained model, continue pre-training it on $d_t$, then fine-tune it on $d_s$. This is done with three random seeds and the performance is averaged over these runs. With this setup, we observe a drastic drop in performance. We hypothesized this is because the amount of in-domain data is insufficient for stable domain adaptation. However, recent work (Röttger and

Pierrehumbert, 2021) has shown that temporal adaptation through continual pre-training even on millions of examples has limited benefit. It should be noted that Röttger and Pierrehumbert (2021) adapt a pre-trained BERT which was pre-trained on recent data overlapping temporally with the data used for the continued pre-training. To completely disentangle the temporal effects of pre-training and assess the effectiveness of continual pre-training, one would also need to pre-train BERT from scratch on older data.

Next, we modify the domain adaptive pre-training by adding an extra fine-tuning step. This method first performs task adaptation, followed by temporal adaptation and then again task adaptation. We start with a pre-trained model, fine-tune it on $d_s$, then pre-train it on $d_t$ and then fine-tune it again on $d_s$. While this method does not improve performance consistently (Table 25), it leads to significant improvement for NER and sentiment for the BERT representation but makes no difference for domain adaptation. For RoBERTa, however, adaptation scores get worse, significantly for NER and the improvement for sentiment analysis is smaller in absolute value than for BERT.

As highlighted in the evaluation setup, multi-test set evaluation is essential for reliable results. In this experiment, if we had evaluated only on 2019 for NER-TTC (numbers omitted here), we would have concluded that this method works well but looking at the summary over different test years, one can see that the change in performance is inconsistent.

### 5.11.2 Self-labeling

Self-labeling has been shown to be an effective technique to detect the need to retrain a model (Elsahar and Gallé, 2019). Here, we explore its use in temporal domain adaptation. We fine-tune a model on $d_s$, use this model to label the data $d_t$ and then use gold-standard $d_s$ and self-labeled $d_t$ to fine-tune another model. The new model is trained on $train_s$ and the full $d_t$ with $dev_s$ as the development set. $d_t$ is weakly labeled (with model predictions) and thus noisy, hence we do not extract a development set from $d_t$ for reliable evaluation. Self-labeling works consistently well, as seen in the results in Table 25, across test years,

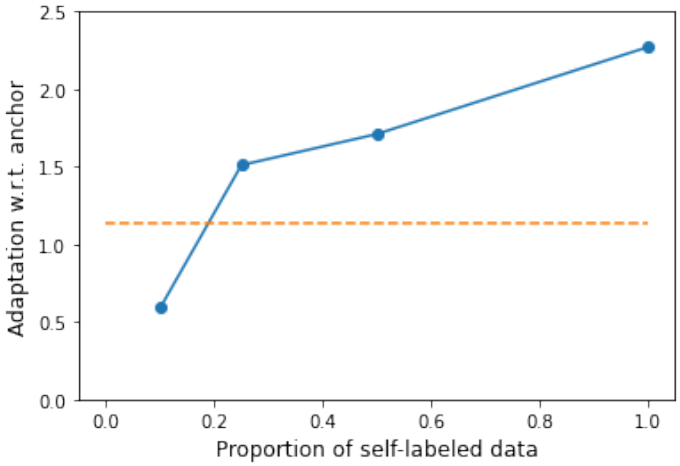



Figure 6: Adaptation score w.r.t. the anchor (2014) by varying the amount of self-labeled data (2015-2018) for NER using BERT. The dashed line shows the adaptation score when just new gold-standard data is used.

representations and tasks[19]. Though adding self-labeled data $d_{t-1}$ does not give the highest reported performance on $d_t$, it improves performance over using just the gold-standard data $d_s$. For NER, F1 improves over using even the $d_t$ gold-standard data[20] (but not over $d_s + d_t$ gold-standard data). For sentiment classification, F1 improves over using just gold-standard $d_s$ but not to the same level as using new gold-standard data for fine-tuning.

Lastly, we explore if continuously adding new self-labeled data further improves performance. All of $d_{s+1}$ to $d_t$ is self-labeled and added to the gold-standard $d_s$. We were able to perform this experiment only for NER since the cumulative data for reviews and domains becomes too large. Adding more data does not improve performance but it does not decrease performance either (numbers omitted), despite the fact that the training data now comprises mainly noisy self-labeled data. More research on optimal data selection with self-labeling is needed. The right data selection may improve performance further.

[19]Adding new data is computationally expensive. For NER, since the amount of data is small because it required actual annotation, we could continue using the same GPU, and just the run time increased. With reviews, we had to upgrade our usage from one to two GPUs in parallel.

[20]For NER using BERT, we vary the amount of self-labeled new data added and observe that with 25% of new self-labeled data, adaptation score exceeds gold-standard fine-tuning (Figure 6.

## 5.12 Experimental Design Recommendations

With this study on the impact of various factors in model training and evaluation that may confound the study of temporal effects, we recommend the following setup for experiments. We highlight the factors that can affect the findings of the study considerably and others that are less important.

1. Evaluate performance on the full grid of possible training and testing time periods. Variation across time periods is considerable and choosing only one can lead to misleading conclusions about changes in performance and utility of methods.

2. Draw development data from the training year and not from the test year to ensure the feasibility of the setup when used in practice.

3. Use multiple input representations since the possibility of improvement via retraining (with labeled or unlabeled data) is representation dependent and we would want an adaptation method that works consistently well.

4. Whenever possible, run experiments without overlap of the time period between the pre-training data and the task data. This will be beneficial for clearing doubts about the reason for performance change. However, such experiments are not necessary since the observed trends seem largely unaffected by such overlap. At a minimum, report the time period for all data used (pre-training, task, external resources).

5. In case of computational constraints, use smaller models. This should not affect trends in the findings. Observed trends are similar across model sizes, even though larger models have better absolute task metrics.

## 5.13 Limitations and Future Work

Work on temporal effects on a variety of tasks, domains and languages is limited by the need to collect a large amount of labeled data. We presented experiments for a range of tasks but focus only on tasks where the answer does not change with time. Other tasks such as question answering and entity linking are time-dependent and are likely to experience deterioration. Two such studies have been performed by Lazaridou et al. (2021); Dhingra et al. (2022) for question answering.

In addition, all of our experiments are on English data. Studying this for other languages, especially those with lower resources, which are most likely to experience deterioration, is harder again due to the need to collect large datasets. Though for multilingual models, one might observe the same trends as English due to transfer learning, experimental evidence will be needed in future work. Additionally, to study adaptation techniques with training on source language or source domain and evaluation on target language or target domain, one would need to match time, domain and task for both, further making such a study harder to execute. Temporal effects are hard to study but future work on different domains and languages will be beneficial, especially in light of our finding that there is not always model deterioration.

Another limitation of our work is due to the need for a large amount of computational resources needed for pre-training from scratch. Röttger and Pierrehumbert (2021) perform extensive experiments on temporal adaptation with continual pre-training but start with a pre-trained BERT which overlaps with task data. Even with the right resources, determining the timestamp for each sentence in the pre-training data is challenging. Wikipedia, a common source of pre-training data, consists of edit histories but there are frequent edits even in the same sentence. If one considers the date when the article was first added, then future data due to edits will get included. Though our experiments hint that task-pre-training data overlap may not impact the results on studies on temporal effects, a clean set of experiments

with no and varying levels of overlap will be essential to understand the effect of such an overlap and motivate the selection of the pre-training data.

Finally, our analyses of the statistical significance of the performance deterioration and adaptation improvement is based on the differences in performance between time periods, for scores averaged across three runs of the model. We report the minimum and maximum adaptation score across runs to account for variation across seeds. However, a single detailed test that takes in account both these variations needs to be designed carefully (Reimers and Gurevych, 2017; Dror et al., 2018). Such analysis will be able to better address questions related to whether it will be more advantageous to update the representations used for the task or to do temporal adaptation. Nevertheless, our work convincingly shows that for individual tasks and representations, deterioration either with respect to an anchor time period or for consecutive time periods is often not statistically significant. Adaptation improvements however are typically significant. This key finding will inform future work.

## 5.14 Conclusion

We presented exhaustive experiments to quantify the temporal effects on model performance. We outline an experimental design that allows us to draw conclusions about both temporal deterioration and the potential for temporal domain adaptation. We find that with pre-trained embeddings, model deterioration is task-dependent and a model need not necessarily experience deterioration for tasks where the label correctness does not depend on time. This finding holds true regardless of whether the pre-training data time period overlaps with the task time period or not. Despite this, temporal adaptation via retraining on new gold-standard data is still beneficial. Therefore, we implemented two methods for temporal domain adaptation without labeling new data. We find that intermediate pre-training is not suitable for temporal adaptation. Self-labeling works well across tasks and representations. This finding motivates future work on how to select data to be labeled and how to maintain a reasonable size for the training data as the continual learning progresses over time.

# CHAPTER 6

# Name vs. Context Learning

*This chapter is based on content originally published in: Oshin Agarwal, Yinfei Yang, Byron C. Wallace and Ani Nenkova, 2021, Interpretability analysis for named entity recognition to understand system predictions and how they can improve. Computational Linguistics, 47(1):117–140. (Agarwal et al., 2021b), and also parts of Oshin Agarwal, Yinfei Yang, Byron C. Wallace and Ani Nenkova, 2020, Entity-switched datasets: An approach to auditing the in-domain robustness of named entity recognition models. arXiv preprint arXiv:2004.04123. (Agarwal et al., 2020). We quantify the extent to which names and contexts are learned by Named Entity Recognition models, by using entity-switched datasets and by developing word-only and context-only models.*

## 6.1 Introduction

The performance of named entity recognition models suffers even within the same genre due to entity, domain and temporal effects. It is natural to ask: What are these models actually learning? Here, we examine the textual clues that lead systems to make predictions. Consider, for instance, the sentence "Nicholas Romanov abdicated the throne in 1917". The correct identification of "Nicholas Romanov" as a person may be due to *(i)* knowing that *Nicholas* is a fairly common name and that *(ii)* the capitalized word after that ending with "-ov" is likely a Slavic last name too. Alternatively, *(iii)* an English language user would know the selectional preferences (Framis, 1994; Akbik et al., 2013; Chersoni et al., 2018) for the subject of the verb abdicate, i.e., that only a person may abdicate the throne. The presence of two words indicates that it cannot be a pronoun, so $X$ in the context "$X$ abdicated the

throne" is most likely a named person.

Such probing of the reasons behind a prediction is in line with early work on NER that emphasized the need to consider both internal (features of the name) and external (context features) evidence when determining the semantic types of named entities (McDonald, 1993). While the impact of names themselves has been explored extensively in prior work (Augenstein et al., 2017b; Derczynski et al., 2017; Fu et al., 2020b), the learning of context is underexplored. We specifically focus on the interplay between learning names as in *(i)*, and recognizing (sentential) contexts with strong selectional preferences as in *(iii)*, given that *(ii)* can be construed as a more general case of *(i)*, in which word shape and morphological features may indicate that a word is a name even if the exact name is never explicitly seen by the system (Table 1 in (Bikel et al., 1999)).

We first use entity-switched datasets, introduced in Chapter 3, to determine the degree of learning names vs. context (i.e. sentence-level contextual clues). Entity-switched datasets consist of the same name in many contexts and many names in the same contexts. Therefore the proportion of contexts in which a name is recognized correctly, and the proportion of names in a context that are recognized correctly, denote name and context learning respectively. We further probe the learning of names and contexts by building models modified to use *only context* or *only word* identities. We quantify the extent to which systems (are able to) exploit word and context evidence respectively. We both these methods, we find that context does inform system predictions, but the major driver of performance is the recognition of certain words as names of a particular type.

## 6.2 Entity-Switched Datasets

Entity-switched datasets are created by replacing named entities in the original text with plausible named entities of the same type while retaining the rest of the text and maintaining its coherence. In a dataset, all occurrences of entities of a given type are replaced by a new target entity (Figure 7a). This results in the same entity in multiple feasible contexts for

| _____ **Samarpita Patnaik** _____ |
| --- |
| Defender **Samarpita Patnaik** rose to intercept a .... |
| **Samarpita Patnaik** is a computer scientist. |
| "We would be very happy with a draw", said **Samarpita Patnaik**. |

(a) Same entity in many contexts

| Defender _____ rose to intercept a ... |
| --- |
| Defender **Samarpita Patnaik** rose to intercept a .... |
| Defender **Hassan Abhas** rose to intercept a .... |
| Defender **Kim Thai** rose to intercept a .... |

(b) Many entities in the same context

Figure 7: Entity–Context combinations generated with entity-switching

that type. The performance of models on the resulting text tells us how well this name was learned by the model. Similarly, for each occurrence of an entity in a dataset, it can be replaced by several entities of the same type (Figure 7b). This setup allows us to identify the templates where errors arose for many names and those where the prediction was accurate, regardless of the name inserted in the template. The proportion of entities recognized correctly for each context then tells us how well this context was learned by the model.

For name learning, an evaluation was performed in Chapter 3. Each occurrence of a person entity in the CoNLL '03 test set was replaced by 340 (15 countries $*$ 20 names) different person names. By reporting the average F1 for entities grouped by their national origin, we found that certain entities, particularly those of Indonesian and Vietnamese origin, are harder to recognize than others. Using the same data, we now determine the extent of context learning. We use only two-word names (318 names out of 340) so as to have the same number of occurrences to compare. There are a total of 2,828 instances of *PER* entities in the test set. Figure 8 provides a histogram of the number of *PER* entities for which the prediction was correct by GloVe word-based model across the 318 names. The prominent peak at the right edge shows that for about half of the contexts, any of the selected names were predicted correctly. These are the contexts that are learned well by the model. Presumably, these are highly predictive contexts that constrain the *PER* semantic type. For about 50 contexts, however, fewer than half of the names were recognized correctly. These contexts were likely not learned at all and the correctness of the prediction depends on the name in the context. A similar-shaped distribution was obtained for the BERT-based model.

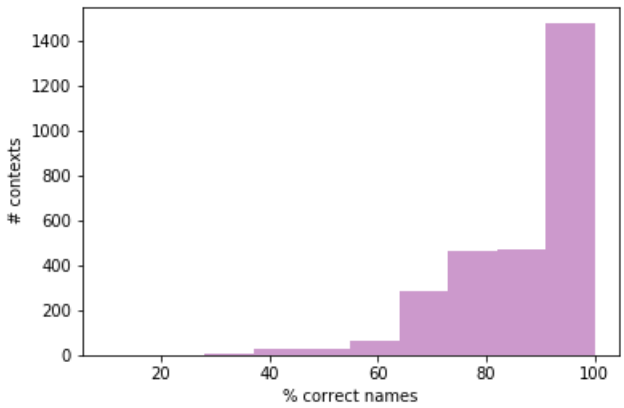



Figure 8: Histogram for % of correct names for each context slot with person names. The total number of contexts is 2828. Almost all names are recognized for about 50% of the contexts, indicating these are the most predictive contexts.

In Table 26 we show the actual sentences (5 for each class), where predictions were most and least accurate. The most learned context is a repetitive context with *PER*, followed by *LOC*, followed by sports scores. These are overly specific to the CoNLL '03 data and would rarely be helpful in general applications of the system. The least predictive contexts are actual sentences and only the more common names are recognized in these. This shows that the system either overfits to a context pattern in the training data or only recognizes the typical names. The specific patterns for BERT predictions are different but point towards the same observation of overfitting to datasets-specific patterns.

## 6.3 Context-only and Word-only Systems

Entity-switched datasets show a difference in the performance of NER models for different words in the same context. The method works for a large-scale evaluation of NER and to draw high-level conclusions about the strong predictive power of word identity. However, we cannot compare precisely the difference in the utilization of the word to its context. Therefore, we now build models to disentangle the performance of systems based on the word identity and the context to determine the extent to which each can inform the model of the entity types. We compare two look-up baselines and several variants of biLSTM-CRF

| Lowest performance | Highest performance |
|---|---|
| 1. Yakomas are hounded in stronghold districts of __ 's Baya people while other tribes have fled areas in rebel hands. | 1. __ (Sweden) 25.76 points |
| 2. German __ in bank nearly gets arrested. | 2. __ (Russia) 23:13.3 |
| 3. " ( __ ) is a convinced advocate of NATO enlargement and of stabilisation of security structures . " | 3. __ (Canada) 1 minute 16.24 seconds |
| 4. Earlier , former Australia test batsman David Boon scored 118 and all-rounder __ hit 113 . | 4. __ (Switzerland) 160.55 |
| 5. Barrick has teamed up with a construction company in the Citra Group of Suharto 's eldest daughter , __ , in what | 5. __ (France) 223.60 |

Table 26: Top patterns for 318 different names with GloVe word-based bi-LSTM-CRF model

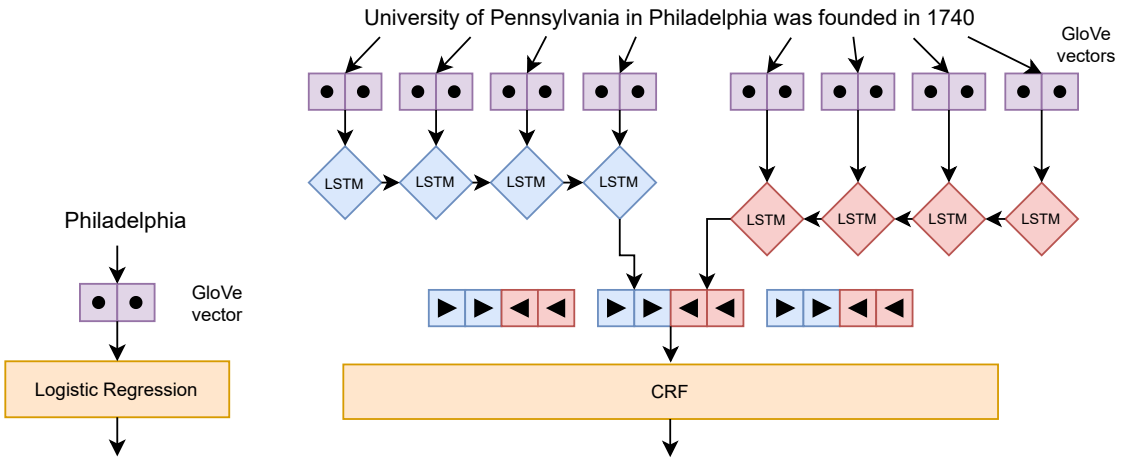
and BERT that use different parts of the input to predict the entity type. The different variants are described below -

**Lookup** Create a table of each word preserving its case and its most frequent tag from the training data. In testing, look up a word in this table and assign its most frequent tag. If the word does not appear in the training data or there is a tie in the tag frequency, mark it as O (outside, not a named entity).

**LogReg** (Figure 9a) Logistic Regression using the GloVe representation of the word only (no context of any kind). This system is equivalent to lookup in both the NER training data and GloVe representations as determined by the data they were trained on.

**GloVe-CRF** Uses GloVe word representations as features in a linear chain CRF. Any word in training or testing that does not have a GloVe representation is assigned a representation equal to the average of all words represented in GloVe. The GloVe embedding parameters are updated during training.

**FWcontext-CRF** This system uses LSTM (Hochreiter and Schmidhuber, 1997) representations only for the text preceding the current word (i.e., run forward from the start to this word), with GloVe as inputs. Here we take the hidden state of the previous word




(a) LogReg (b) Bicontext-CRF

Figure 9: Glove-based word-only (LogReg) and context-only (Bicontext-CRF) models

as the representation of the current word. This incorporates non-local information not available to the two previous systems, from the part of the sentence before the word.

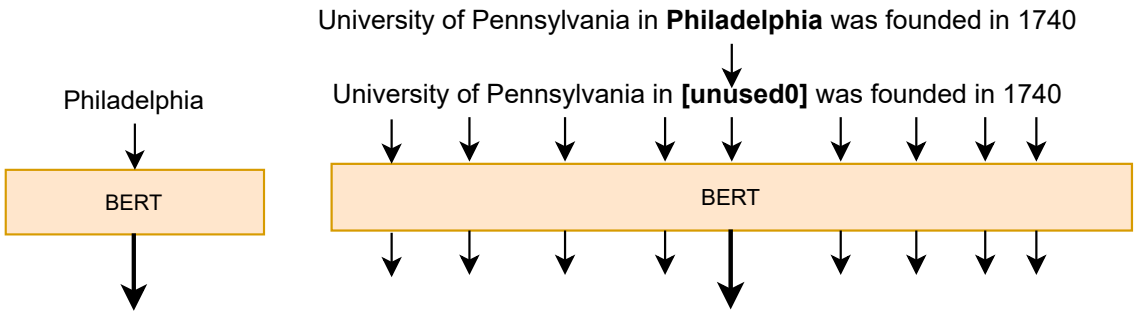
**BWcontext-CRF** Backward context-only LSTM with GloVe as inputs. Here we reverse the sentence sequence and take the hidden state of the next word in the original sequence as the output representation of the current word.

**BIcontext-CRF** (Figure 9b) Bidirectional context-only LSTM with GloVe as input. The forward and backward context-only representations are concatenated taking the hidden state as in the two systems above and not the hidden state of the current word.

**BIcontext-word-CRF** (Huang et al., 2015) The feature representing the word is the hidden state of the LSTM after incorporating the current word. The forward and backward representations are concatenated.

We use 300d cased GloVe (Pennington et al., 2014b) vectors trained on Common Crawl. The models are trained for 10 epochs with a learning rate of 0.01 and weight decay of 1e-4. A dropout of 0.5 is applied on the embeddings and the LSTM hidden layer dimension is 100.

**Full BERT** We use the original public large cased model and apply the default fine-tuning




(a) Word-only BERT (b) Context-only BERT

Figure 10: BERT-based word-only and context-only models

strategy. We use the implementation in (Wolf et al., 2020) and finetune the model for 3 epochs with a learning rate of 5e-06 and a maximum sequence length of 128.

**Word-only BERT** (Figure 10a) Fine-tuned using only the word as the input resulting in a word-only representation. Each word token can appear multiple times with various entity types in the training data and the frequency for each is maintained in the training data. This is an unusual setting since BERT is pre-trained to predict the word based on the context. However, in this case there may not be any context unless the word is tokenized into sub-words. Nonetheless, the model performs very well even in this setting since fine-tuning uses both the word and the context.

**Context-only BERT** (Figure 10b) Since decomposition of BERT representations into word-only and context-only is not straightforward,[21] we adopt an alternate strategy to test how BERT fairs without seeing the word itself. We use a reserved token from the vocabulary '[unused0]' as a mask for the input token so that the system is forced to make decisions based on the context and does not have a prior entity type bias associated with the mask. We do this for the entire dataset, masking one word at a time. It is important to note that the word is only masked during testing and not during fine-tuning.

[21] We tried a few techniques such as projecting the BERT representations into word and context spaces by learning to predict the word itself and the context words as two joint tasks, but this did not work well.

| System | CoNLL | | | Wikipedia | | | MUC-6 | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| *Full system* | | | | | | | | | |
| BIcontext-word-CRF | 90.7 | 91.3 | 91.0 | 66.6 | 60.8 | 63.6 | 90.1 | 91.8 | 90.9 |
| *Words only* | | | | | | | | | |
| Lookup | 84.1 | 56.6 | 67.7 | 66.3 | 28.5 | 39.8 | 81.4 | 54.4 | 65.2 |
| LogReg | 80.2 | 74.3 | **77.2** | 58.8 | 48.9 | **53.4** | 75.1 | 71.7 | **73.4** |
| *Words + local context* | | | | | | | | | |
| Glove-CRF | 80.8 | 77.3 | **79.0** | 63.3 | 45.8 | **53.1** | 82.1 | 77.0 | **79.5** |
| *Non-local context only* | | | | | | | | | |
| FWcontext-CRF | 71.3 | 39.4 | 50.8 | 53.3 | 19.3 | 28.4 | 71.9 | 58.9 | **64.7** |
| BWcontext-CRF | 69.5 | 47.7 | 56.6 | 46.6 | 21.7 | 29.6 | 74.0 | 49.4 | 59.2 |
| BIcontext-CRF | 70.1 | 52.1 | **59.8** | 51.2 | 21.4 | **30.2** | 66.4 | 56.5 | 61.1 |
| *BERT* | | | | | | | | | |
| Full | 91.9 | 93.1 | 92.5 | 75.4 | 75.1 | 75.2 | 96.1 | 97.2 | 96.7 |
| Word-only | 80.0 | 80.5 | **80.3** | 61.6 | 54.8 | **58.0** | 77.9 | 75.3 | **76.5** |
| Context-only | 43.1 | 64.1 | 51.6 | 39.7 | 76.2 | 52.2 | 75.6 | 71.6 | 73.5 |

Table 27: Performance of GloVe word-level BiLSTM-CRF and BERT. All rows are for the former and only the last two rows for BERT. Local context refers to high precision constraints due to sequential CRF. Non-local context refers to the entire sentence. No document level context is included. The first two panels were trained on the Original English CoNLL 03 training data and tested on the original English CoNLL 03 test data and the WikiGold data. The last panel was trained and tested on the respective splits of MUC-6. Highest F1 in each panel is boldfaced, excluding the full systems.

### 6.3.1 Results

We evaluate the models on CoNLL '03 and MUC-6. Our goal is to quantify how well the models can work if the identity of the word is not available and to compare that to situations in which *only* the word identity is known. Additionally, we evaluate the systems trained on CoNLL data on Wikigold (Balasuriya et al., 2009). We use the IO labeling scheme and evaluate the systems via micro-F1, at the token level, reporting results in Table 27. The first line in the table (BIcontext-word-CRF) corresponds to the model in Huang et al. (2015).

**Word only systems** The results in the *Word only* rows are as expected: We observe low recall and high precision. The results are consistent across CoNLL as well as MUC6 datasets. On the cross-domain evaluation, however, when the system trained on CoNLL

| | % PER with honorifics | % ORG in sports scores |
|---|---|---|
| CoNLL train | 2.74 | 25.89 |
| CoNLL test | 2.59 | 32.37 |
| Wikipedia | 4.65 | 0.00 |
| MUC-6 train | 27.46 | 0.00 |
| MUC-6 test | 35.08 | 0.00 |

Table 28: Repetitive context patterns in the datasets. In CoNLL, several organizations occur in sports scores. In MUC-6, several person names are preceded by an honorific.

is evaluated on Wikigold, the recall drops considerably. This behavior may be attributed to the dataset: Many of the entities in the CoNLL training data also appear in testing, a known undesirable fact (Augenstein et al., 2017b). We find that 64.60% and 64.08% of entity tokens in CoNLL and MUC6 test set are seen in the respective training sets. However, only 41.55% of entity tokens in Wikigold are found in the CoNLL training set. The use of word representations (*LogReg* row) contributes substantially to system performance, especially for the MUC-6 dataset in which few names appear in both train and test. Given the impact of the word representations, it would seem important to track how the choice and size of data for training the word representations influence system performance.

**Word and local context combinations** Next, we consider the systems in the *Word + local context* rows. CRFs help to recognize high precision entity-type local contextual constraints, e.g., force a *LOC* in the pattern '*ORG ORG LOC*' to be an *ORG* as well. Another type of high-precision constraining context is word-identity based, similar to the information extraction work discussed above, and constrains X in the pattern 'X said' to be *PER*. Both of these context types were used in Liao and Veeramachaneni (2009) for semi-supervised NER. The observed improved precision and recall of *GloVe-CRF* over *LogReg* indicates that the CRF layer modestly improves performance.

**Context features only** We compare *Context only* systems with non-local context. In CoNLL, the context after a word appears to be more predictive, while in MUC-6 the forward context is more predictive. In CoNLL, some of the right contexts are too corpus specific,

| | Full | Word-only | Ctx-only |
|---|---|---|---|
| Original | 91.76 | 71.44 | 38.00 |
| India | 82.63 | 43.99 | 32.16 |
| Vietnam | 77.08 | 26.15 | 23.53 |

Table 29: Span F1 of full, word-only and context-only BERT on all entity types in the CoNLL '03 test data.

such as 'X 0 Y 1' being predictive of X and Y as organizations with the example occurring in reports of sports games, such as 'France 0 Italy 1'. 32.37% of the *ORG* entities in CoNLL test split occur in such sentences. MUC-6, on the other hand, contains many examples that include honorifics, such as 'Mr. X'. 35.08% of *PER* entities in MUC-6 test split are preceded by an honorific, whereas in CoNLL and MUC6, this is the case only for 2.5% and 4.6% entities, respectively. Statistics on these two patterns are shown in Table 28. We compare *Context only* systems with non-local context. In CoNLL data, the context after a word appears to be more predictive, while in MUC-6 the forward context is more predictive. In CoNLL, some of the right contexts are too corpus specific, such as 'X 0 Y 1' being predictive of X and Y as organizations with the example occurring in reports of sports games, such as 'France 0 Italy 1'. 32.37% of the *ORG* entities in CoNLL test split occur in such sentences. MUC-6, on the other hand, contains many examples that include honorifics, such as 'Mr. X'. 35.08% of *PER* entities in MUC-6 test split are preceded by an honorific, whereas in CoNLL and MUC6, this is the case only for 2.5% and 4.6% entities, respectively. Statistics on these two patterns are shown in Table **??**. We provide the list of honorifics and regular expressions for sports scores used to calculate this in Appendix B.

Clearly, systems with access to only word identity perform better than those with access to only the context (drop of ∼20 F1 in all three datasets). It brings up an important question that we will answer in the next chapter through human evaluation — are systems unable to utilize context well or are many contexts not predictive of the entity types? Moreover, while one would have expected that the context features have high precision and low recall, this is indeed not the case: the precision of the BIcontext-CRF system is consistently lower

| | Train Genre | | | | | |
|---|---|---|---|---|---|---|
| Test Genre | nw | bn | bc | tc | mz | wb |
| | | | Full | | | |
| nw | 92.8 | 85.8 | 78.7 | 52.5 | 82.0 | 65.8 |
| bn | 88.5 | 91.3 | 83.9 | 58.1 | 81.8 | 59.8 |
| bc | 80.7 | 81.5 | 84.1 | 55.5 | 76.3 | 53.4 |
| tc | 73.8 | 76.6 | 68.5 | 69.7 | 68.8 | 48.2 |
| mz | 86.4 | 82.4 | 82.7 | 42.3 | 86.7 | 68.3 |
| wb | 49.0 | 47.9 | 46.4 | 42.3 | 48.3 | 71.1 |
| | | | Word | | | |
| nw | 52.0 | 43.0 | 38.8 | 34.1 | 42.9 | 10.5 |
| bn | 60.3 | 66.3 | 59.7 | 54.4 | 59.4 | 17.0 |
| bc | 43.9 | 49.0 | 45.7 | 37.0 | 40.5 | 13.0 |
| tc | 38.4 | 38.2 | 46.4 | 50.9 | 38.3 | 10.7 |
| mz | 42.9 | 46.8 | 45.1 | 40.9 | 53.4 | 9.9 |
| wb | 31.3 | 31.5 | 29.8 | 30.9 | 31.6 | 23.2 |
| | | | Context | | | |
| nw | 20.2 | 20.7 | 15.8 | 7.4 | 17.7 | 10.2 |
| bn | 21.2 | 23.4 | 19.9 | 4.4 | 19.9 | 11.0 |
| bc | 14.0 | 14.8 | 14.4 | 3.3 | 14.8 | 5.9 |
| tc | 10.2 | 8.6 | 7.1 | 2.1 | 9.8 | 4.0 |
| mz | 11.8 | 17.0 | 11.3 | 2.3 | 19.5 | 9.9 |
| wb | 11.0 | 10.9 | 9.6 | 5.6 | 9.4 | 9.5 |

Table 30: Span F1 of full, word-only and context-only BERT on all entity types in the Ontonotes test data.

than the precision for the full system and the logistic regression model. This means that a better system will not only learn to recognize more contexts but also would be able to override contextual predictions based on features of the word in that context.

**Contextualized word representations** To recap, word-only BERT is finetuned using only one word at a time without any context. Full BERT and context-only BERT are the same models finetuned on the original dataset and differ only in inference. For context-only BERT, a reserved vocabulary token '[unused0]' is used to hide one word at a time to get a context-only prediction. Full BERT improves over the GloVe-based BiLSTM-CRF as expected. Word-only BERT performs better than the context-only BERT but the difference in performance is not as pronounced as in the case of the GloVe-based BiLSTM-CRF, except on the CoNLL corpus due to the huge overlap of the training and testing set entities as noted

| Test Sub-domain | Train Sub-domain | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | a | b | c | e | f | m | n | s | o |
| **Full** | | | | | | | | | |
| arts | 92.1 | 88.7 | 78.4 | 87.9 | 87.1 | 89.7 | 88.7 | 85.7 | 89.2 |
| business | 91.6 | 95.7 | 72.0 | 90.2 | 90.0 | 93.2 | 91.9 | 84.7 | 92.0 |
| classified | 73.2 | 78.1 | 94.7 | 90.4 | 67.1 | 84.7 | 83.6 | 68.8 | 75.3 |
| editorial | 91.0 | 94.0 | 67.0 | 96.4 | 92.1 | 94.3 | 94.6 | 82.1 | 94.1 |
| foreign | 90.4 | 92.9 | 64.2 | 92.1 | 96.9 | 93.3 | 93.9 | 80.0 | 93.1 |
| metropolitan | 90.4 | 91.0 | 78.4 | 91.4 | 90.6 | 95.0 | 92.8 | 86.2 | 92.5 |
| national | 90.3 | 94.0 | 79.8 | 94.2 | 93.6 | 94.9 | 96.2 | 86.6 | 94.1 |
| sports | 84.6 | 81.7 | 78.3 | 80.4 | 78.1 | 84.0 | 77.9 | 94.8 | 82.9 |
| others | 90.1 | 89.0 | 76.7 | 88.9 | 87.6 | 90.8 | 90.3 | 85.5 | 92.0 |
| **Word** | | | | | | | | | |
| arts | 75.8 | 64.6 | 66.7 | 66.5 | 66.0 | 71.1 | 69.5 | 63.7 | 71.2 |
| business | 69.3 | 74.0 | 61.3 | 65.0 | 65.6 | 70.4 | 70.3 | 60.2 | 68.7 |
| classified | 77.0 | 70.2 | 83.7 | 79.6 | 58.6 | 75.6 | 72.1 | 68.0 | 76.3 |
| editorial | 76.5 | 73.2 | 71.9 | 79.1 | 76.2 | 80.4 | 78.5 | 65.3 | 79.9 |
| foreign | 74.4 | 72.3 | 69.5 | 72.0 | 85.9 | 77.2 | 76.9 | 70.2 | 77.3 |
| metropolitan | 69.9 | 59.1 | 58.7 | 69.9 | 61.8 | 80.6 | 71.4 | 54.7 | 73.2 |
| national | 76.5 | 74.0 | 72.3 | 76.9 | 76.6 | 79.9 | 83.3 | 61.5 | 78.7 |
| sports | 66.7 | 65.2 | 64.4 | 62.4 | 64.2 | 69.5 | 67.7 | 85.0 | 65.1 |
| others | 74.5 | 66.0 | 69.1 | 69.4 | 69.7 | 76.0 | 74.4 | 62.5 | 78.9 |
| **Context** | | | | | | | | | |
| arts | 39.0 | 24.9 | 16.0 | 34.9 | 35.3 | 37.8 | 37.8 | 32.1 | 32.1 |
| business | 32.6 | 22.8 | 12.8 | 30.7 | 25.5 | 31.7 | 31.8 | 24.2 | 24.2 |
| classified | 17.4 | 17.2 | 44.1 | 18.4 | 16.5 | 18.6 | 17.1 | 13.9 | 13.9 |
| editorial | 26.9 | 22.1 | 12.0 | 27.3 | 21.1 | 25.4 | 25.5 | 19.1 | 19.1 |
| foreign | 29.9 | 21.6 | 12.8 | 30.9 | 27.3 | 30.6 | 31.0 | 15.8 | 25.5 |
| metropolitan | 38.5 | 23.5 | 15.2 | 35.2 | 34.3 | 40.7 | 39.3 | 33.5 | 33.5 |
| national | 40.3 | 28.4 | 14.8 | 39.2 | 36.6 | 40.9 | 42.3 | 34.1 | 34.1 |
| sports | 28.0 | 21.4 | 14.7 | 25.4 | 18.9 | 21.1 | 16.5 | 11.3 | 11.3 |
| others | 35.2 | 23.7 | 16.9 | 32.7 | 29.5 | 32.9 | 33.5 | 26.6 | 26.6 |

Table 31: Span F1 of full, word-only and context-only BERT on all entity types in the NYT test data.

earlier. We also note that context-only BERT performs better or worse than context-only BiLSTM, depending on the corpus. These results show BERT is not always better at capturing contextual clues. While it is better in certain cases, it also misses these clues in some instances for which the BiLSTM makes a correct prediction.

We examine the performance of BERT on a sample of sentences for which BiLSTM context representations are sufficient to make a correct prediction and a sample of sentences where it is not. We randomly sampled 200 examples from CoNLL 03 where the context-only

| | Train Year | | | | |
|---|---|---|---|---|---|
| Test Year | 2014 | 2015 | 2016 | 2017 | 2018 |
| | | | Full | | |
| 2015 | 64.69 | - | - | - | - |
| 2016 | 65.60 | 66.89 | - | - | - |
| 2017 | 65.90 | 65.62 | 66.68 | - | - |
| 2018 | 64.08 | 64.59 | 65.76 | 65.17 | - |
| 2019 | 71.70 | 73.68 | 73.06 | 74.05 | 76.20 |
| | | | Word | | |
| 2015 | 45.78 | - | - | - | - |
| 2016 | 46.26 | 50.19 | - | - | - |
| 2017 | 45.61 | 46.76 | 47.41 | - | - |
| 2018 | 43.30 | 45.45 | 46.44 | 46.37 | - |
| 2019 | 46.13 | 45.14 | 46.50 | 49.50 | 51.50 |
| | | | Context | | |
| 2015 | 8.05 | - | - | - | - |
| 2016 | 5.22 | 7.95 | - | - | - |
| 2017 | 5.90 | 9.57 | 7.94 | - | - |
| 2018 | 9.73 | 14.39 | 11.45 | 21.34 | - |
| 2019 | 11.04 | 15.72 | 14.40 | 25.70 | 22.72 |

Table 32: Span F1 of full, word-only and context-only BERT on all entity types in the TTC test data.

LSTM was correct (Sample-C) and another 200 where it was incorrect (Sample-I). Context-only BERT is correct on 71.5% examples in Sample-C but fails to make the correct prediction on the remaining about 28.5% that are easy for the BiLSTM context representation. In contrast, it is also able to correctly recognize the entity type in 53.22% of the cases in Sample-I, where BiLSTM context is insufficient for prediction. The instances for which context BiLSTM is correct are not a subset of those correct using context BERT. Since the context-only models are inconsistent, the question remains what contextual clues are both learning and are those consistent with human judgments.

Lastly, as with the entity-switching experiments, to be consistent with the rest of the thesis, we build word-only and context-only variants of bert-large-cased and report span-level F1 on CoNLL '03, all domains of Ontonotes, New York Times sub-domains (from chapter 4) and Temporal Twitter Corpus (from Chapter 5). We train three models with different seeds and report the average F1. Hyperparameter details can be found in the

appendix. The results are shown in Tables 29, 30, 31 and 32. The trends are consistent with the results reported above. Word-only models perform very well on all the datasets. In comparison, the context-only models are much worse off, and in several cases have extremely poor performance.

## 6.4 Conclusion

We zeroed in on the question of interpretability of named entity recognition systems, specifically examining the degree of name and context learning using two methods. First, we used entity-switched datasets to determine which contexts are learned by the models. We found that contexts learned best are dataset-specific patterns and those with simple predictive clues are often missed. Next, we examined models to attribute the degree of performance to the word and its context. We built models that use only the word or only the context to make predictions. We found that current systems, including those built on top of contextualized word representations, pay more attention to the current word than to the contextual features. The question remains whether the contextual clues are insufficiently predictive i.e. they are ambiguous, or if they are just not being learned by the models. The answer to this question will determine if we can build models that can utilize context any better.

# CHAPTER 7

# Constraining Contexts

*Part of this chapter is based on content originally published in: Oshin Agarwal, Yinfei Yang, Byron C. Wallace and Ani Nenkova, 2021, Interpretability analysis for named entity recognition to understand system predictions and how they can improve. Computational Linguistics, 47(1):117–140. (Agarwal et al., 2021b). Through a human study, we determine the feasibility of developing Named Entity Recognition models that can focus more on context and can recognize those with strong selection preferences for the entity type. As a first step in this direction, we propose two methods for automatically identifying such contexts. Based on the results of both the human study and the two methods, we find that developing such models is hard. We instead use these methods to expand the set of labels plausible in a given context and use them to perform training data augmentation with perturbations without any external resources.*

## 7.1 Introduction

Named Entity Recognition (NER) models with access to just the context of a word are not able to determine the class of the entity (or non-entity) word well. The poor performance of the context-only models would naturally prompt us to assume that models are unable to utilize contextual clues well and set about building models that focus more on context. However, it is worth taking a step back to analyse entity contexts to determine whether they have sufficiently strong clues to constrain the entity type.

In the sentence "Dr. X is a computer scientist.", we can say with confidence that X is a person's name. The context imposes a selectional restriction on the semantic type of words

that can take the place of X. Similarly, in "X Romanov abdicated the throne in 1917", due to strong selectional preferences of the word 'abdicate' for the semantic class of the subject and the presence of the last name 'Romanov', X should be most likely a person's name. However, X can also sometimes be the non-entity word 'King' and the identity of X is necessary for the final disambiguation. The context in this case imposes a strong selectional preference rather than restriction for the semantic class of X. In contrast to these examples, in "X won the match", X can be the name of a person or a team or even a pronoun (non-entity). All the types seem equally likely. The context is highly ambiguous and therefore imposes no (or weak) selectional preferences for the type of X. While it is possible that one of the entity classes is more likely, as stated in Resnik (1993), selectional preferences are theoretical probabilities in the mind of a competent user of the given language. For practical purposes of building useful systems, they can be approximated by empirical probabilities based on a large corpus.

The task of named entity recognition is then to learn these empirical probabilities. The full model which accesses both the word and its context takes into account the identity of the word for the final disambiguation. The context-only model learns the type probabilities for an unknown word X in a given context. For a context that imposes selectional restrictions or strong selectional preferences, the context-only model, and also the full model, should be correctly able to recognize the entity type based solely on the context, regardless of the word in it. For all other cases, we can not expect a model to always recognize the entity correctly, even when it is given access to the word. The accuracy would depend on whether the word has been seen by the model in the (pre-)training data.

We define constraining contexts as the contexts that impose selectional restrictions or strong selectional preferences on the semantic class of the target word in it. We consider only sentence-level context since most models work at the sentence level. Below are some examples of constraining contexts for different entity types. The type of X in each of these contexts should generally be the same, irrespective of X.

**PER** My name is $X$.

**LOC** The flight to $X$ leaves in two hours.

**ORG** The CEO of $X$ resigned.

In the first example, X is most likely a person's name. Though a non-entity word like 'unique' is possible, it is much less likely. Similarly, in the second example, a non-entity work like 'hell' is possible but it is much more likely to be a destination. In the last example, one can imagine X to be again 'hell' in a snide remark, but in all probability, it refers to an organization.

Constraining contexts (or any context) can be for a word or an entity span. While span-level is the more natural choice, popular NER models operate at the word level, therefore for simplicity, we work with words as well. Constraining contexts can also be general or domain/dataset-specific. In the examples above, we can make judgments about the selectional preferences without any other information, therefore these are general constraining contexts. CoNLL '03 dataset contains several sports scores listed in a specific pattern. Only by knowing that the sentences are sports scores or derived from CoNLL can we judge the entity type solely from the context. These are domain-specific constraining contexts.

Our initial question of whether contexts have sufficiently strong clues can now be rephrased as to how often are contexts constraining. To answer this question, we perform a human study. Specifically, we task human annotators with inferring entity types using only (sentential) context. The goal of this study can be viewed in two ways–

1. **Feasibility of More Context Utilization**: Can humans identify entity types correctly when the model is incorrect? If yes, then there is scope for better context utilization. If not, then we have reached the peak of context utilization.

2. **Feasibility of Model for Better and Realistic Generaliation**: Are contexts usually constraining or ambiguous? If a significant portion is constraining, then building

a model that identifies them should be the next step for better and realistic generalization. However, if they are generally ambiguous, then depending on the frequency and the scope for improvement, building such a model might not be a meaningful effort.

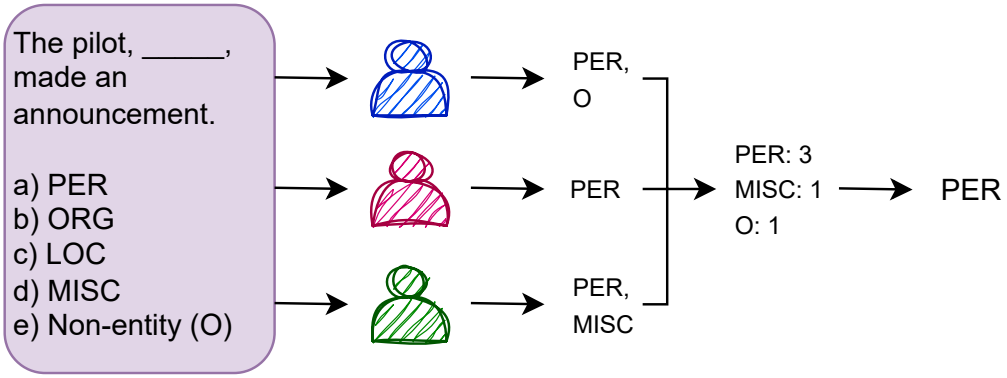
In this chapter, we describe the human study including both its setup and results. The difficulty of the task and human accuracy depends on how the task is designed. When asked to give example words of each plausible type instead of merely selecting the types, humans perform much better on the task. The task is hard even for humans but annotators are still able to correctly identify the entity type from the context alone in several cases. The strength of the preference varies and can be determined by looking at the distribution of responses. About 50% of the instances used for the study were constraining as per human judgments. Therefore, we explore some methods for automatically recognizing constraining contexts, namely context-only models, and k-nearest neighbours with words predicted by masked language models. These methods work fairly well and are able to recognize several constraining contexts. However, incorporating them in models remains a challenging and open problem.

## 7.2 Feasibility of More Context Utilization

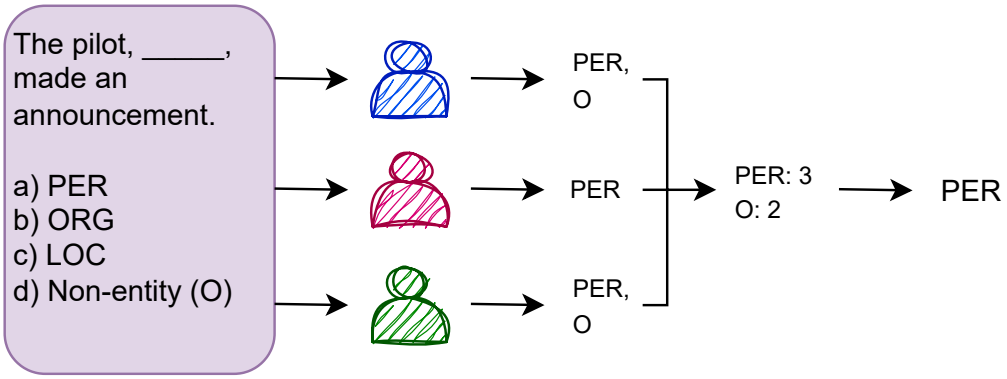
We perform a study with humans, assessing the feasibility of inferring entity types using only contextual cues with the word identity. Only sentence-level context is used as all models operate at the sentence level.

### 7.2.1 Pilot Study

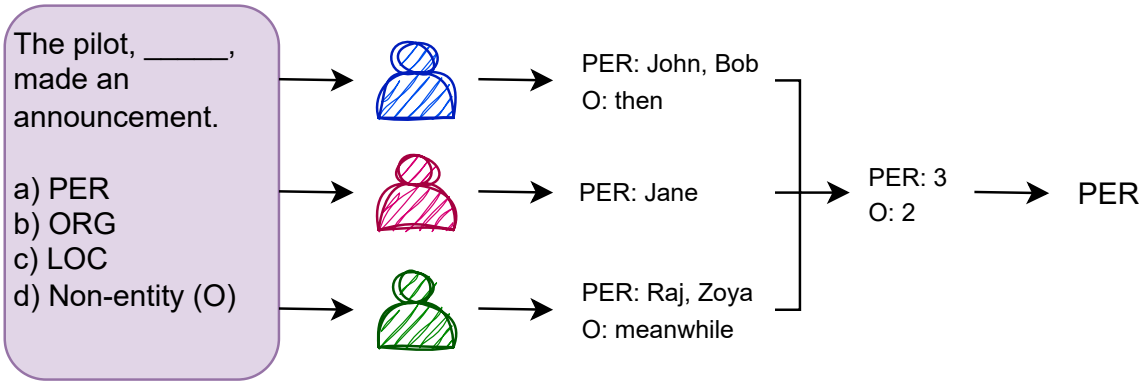
Typical NER annotation consists of marking all entities in a given sentence. For our study, each instance consists of a sentence with a target word which will be blanked/masked. We ask three annotators to determine the type of the missing target word. The types are provided as a list and multiple types can be selected. Instances are presented in batches of 10. To ensure the quality of the annotation, one example is repeated in every batch and




(a) Pilot Study Design




(b) Simplified Study Design




(c) Final Study Design

Figure 11: Three different designs of the human study to determine entity types based solely on the entity context.

| Sentence | Word | Label | Human |
|---|---|---|---|
| Lang said he ___ conditions proposed by Britain's Office of Fair Trading, which was asked to examine the case last month. | supported | O | - |
| ___ Vigo 15 5 5 5 17 17 20 | Celta | ORG | O |
| The years I spent as manager of the Republic of ___ were the best years of my life. | Ireland | LOC | - |

Table 33: Human errors when the context-only system was correct but humans incorrect.

the answers of the annotators that are inconsistent on this repeated example are discarded. Furthermore, we include an example either directly from the instructions or very similar to an example in the instructions. If an annotator does not select the type from the example at a minimum, we assume that they either did not read the instructions carefully, or that they did not understand the task, and we discard their answer as well. When responses are discarded, the batch is republished to get a total of three annotators per batch. Since humans are allowed to select multiple options, we do not expect them to fully agree on all of the options. The goal is to select the most likely option and so we take the label with the most votes i.e. the majority label as the final human label.

We first select 20 instances from CoNLL on which the context-only BiLSTM model was correct and another 200 instances for which the context-only BiLSTM was incorrect. The sample from correct prediction is smaller because we see overwhelming evidence that whenever the BiLSTM prediction is correct, humans can also easily infer the correct entity type from contextual features alone. The label options available to humans are the same as the CoNLL label set (*PER*, *LOC*, *ORG*, *MISC*, *O*). The task design is illustrated in Figure 11a. For 85% of the 20 instances, the majority label provided by the annotators was the same as the true (and predicted) label. Table 33 shows the three (out of 20) examples in which humans did not agree on the category or made a wrong guess. These 20 instances serve as a sanity check as well as a check for annotator quality for the examples where the system made incorrect predictions.

For the 200 instances where the context-only BiLSTM-CRF model made incorrect pre-

| Sentence | Word | Label | Human | GloVe | BERT |
|---|---|---|---|---|---|
| Analysts said the government , while anxious about ___ 's debts , is highly unlikely to bring the nickel , copper , cobalt , platinum and platinum group metals producer to its knees or take measures that could significantly affect output . | Norilisk | ORG | ✓ | O | O |
| 6. Germany III ( Dirk Wiese , Jakobs ___ ) 1:46.02 | Marco | PER | ✓ | O | O |
| - Gulf ___ Mexico : | of | LOC | ✓ | MISC | O |
| About 200 Burmese students marched briefly from troubled Yangon ___ of Technology in northern Rangoon on Friday towards the University of Yangon six km ( four miles ) away , and returned to their campus , witnesses said . | Institute | ORG | ✓ | O | LOC |
| NOTE - Sangetsu Co ___ is a trader specialising in interiors . | Ltd | ORG | ✓ | O | ✓ |
| Russ Berrie and Co Inc said on Friday that A. ___ Cooke will retire as president and chief operating officer effective July 1 , 1997 . | Curts | PER | ✓ | ORG | ✓ |
| ASEAN groups Brunei , Indonesia , Malaysia , the ___ , Singapore , Thailand and Vietnam . | Philippines | LOC | ✓ | O | ✓ |
| Their other marksmen were Brazilian defender Vampeta ___ Belgian striker Luc Nilis , his 14th of the season . | and | O | PER | PER | ✓ |
| On Monday and Tuesday , students from the YIT and the university launched street protests against what they called unfair handling by police of a brawl between some of their colleagues and restaurant owners in ___ . | October | O | LOC | LOC | LOC |
| The longest wait to load on the West ___ was 13 days . | Coast | O | MISC | LOC | LOC |
| , 41 , was put to death in ___ 's electric chair Friday . | Florida | LOC | - | O | ✓ |
| Wall Street , since the bid , has speculated that any deal between Newmont and ___ Fe would be a "bear hug , " or a reluctantly negotiated agreement where the buyer is not necessarily a friendly suitor . | Santa | ORG | - | LOC | LOC |

Table 34: Examples of human evaluation.

dictions, we received a variety of responses. The human annotators were able to correctly determine the label for 24% (48) of the sentences where the context-only BiLSTM-CRF made errors, indicating some room for improvement in a context-only system. For the remaining 76% instances, humans could not predict the entity type from only the context. For 55.5% of the cases, there was a human majority label but it was not the same as the true label. This means that the label of the word in the datasets was contradictory to the most likely as determined by the context. We will discuss more about such selectional preferences in the next section.

Remarkably, BERT has correct as well as incorrect predictions on the examples from both classes of examples where humans were correct (class 1) or not (class 2). It was correctly able to determine the entity type for 65.9% of cases in class 1 and 49.3% of cases in class 2. Examples of each are shown in Table 34. These results show that neither system is learning the same contextual clues as humans.

### 7.2.2 Simplified Task

Humans could correctly identify the type without seeing the target word for less than a quarter of the errors made by the BiLSTM-CRF. The task was hard and we therefore simply it (Figure 11b) by making three changes–

1. On Mechanical Turk, annotators spend only a few minutes attempting each task. Therefore classes like miscellaneous can be difficult to understand. We remove MISC as a separate label in the list of options and treat words marked as MISC in the dataset as O.

2. We present only five examples per batch instead of ten but pay the same per batch as above. This may encourage annotators to spend more time on the task.

3. We saw that the task was hard and therefore expecting exact consistency on the repeated example might be an unrealistic bar for quality control. We simplify this by

simply adding a test example to the batch, asking annotators to select a specific label from the options. This will likely eliminate spammers but keep annotators who would have attempted the task sincerely but still have been inconsistent on the repetition.

With the above changes, we conduct the study again on the 200 instances. This time, humans were correctly able to determine the type for 42.5% of the instances. This is a huge jump over the 24% accuracy in the pilot study.

### 7.2.3 Final Redesigned Task

The redesign of the study simplified the task, allowing annotators to achieve much higher accuracy on context-only NER. Part of the instructions in the task asked annotators to think of appropriate words for each label and then select each label that was possible for the target blanked word in the sentence. The process was implicitly meant to nudge the annotators in the right direction but not increase the task difficulty by asking them to write the words. However, this may lead to annotators thinking less carefully. Therefore, we made this an explicit part of the task. We asked annotators to write one or more possible words for each possible label instead of just selecting the label. If they were certain that a label was possible but was unable to think of a word, they could write the word 'UNKNOWN'.[22]

With this redesign, we again perform the study with the same 200 instances from CoNLL. The human accuracy further jumped to 52% with this change. The task is indeed difficult but a careful design makes a big difference. The human accuracy for the three versions of the task is shown in Table 35 for comparison. For the pilot study, we also report the accuracy by treating MISC as O so as to have to the same number of classes for comparison.

[22]This option was rarely utilized by any annotator.

| Task | Correct (%) |
|---|---|
| Pilot | 24.0 |
| Pilot (MISC -> O) | 28.5 |
| Simplified | 42.5 |
| Final Redesigned | 52.0 |

Table 35: Human accuracy w.r.t. dataset labels on the 200 instances from CoNLL.

### 7.2.4 Expanding Datasets

We adopt the last version of the study which involves writing words and expand it to more datasets. We randomly select more sentences from CoNLL '03 (regardless of correct/incorrect), all genres of Ontonotes (except telephone conversation) and WNUT '17. We then mask a mix of entity and non-entity words in these sentences and perform the same study. The number of instances in each dataset and the percentage correctly identified by humans based on the context is shown in Table 36. About 70-80% types can be correctly identified in each case. This number is higher than the 52% on the 200 instances from CoNLL because those were selected where context-only BiLSTM was incorrect so not reflective of general ability, but rather meant to ascertain if humans can do better than the model. The percentage of instances that can be correctly identified by context-BERT is close to the percentage identified correctly by humans, except for Twitter. However, all the instances recognized correctly by BERT are not a subset of those recognized by humans, as observed in the pilot study. Therefore, there are a significant proportion of instances recognized correctly by humans but not by systems. There is room for better utilization of contextual clues.

## 7.3 Feasibility of Model for Better and Realistic Generalization

In the last section, we determined the accuracy of humans in determining the entity type solely based on the context. However, an incorrect human label does not translate to the lack of contextual clues in these instances. Contextual clues may exist but depending on the

| Dataset | # instances | Correct (%) | | | |
|---|---|---|---|---|---|
| | | Humans | ctx bilstm-crf | ctx BERT | Human+ctx-BERT |
| CoNLL-20 | 20 | 80.0 | 100.0 | 85.0 | 70.0 |
| CoNLL-200 | 200 | 52.0 | 0.0 | 54.5 | 41.0 |
| CoNLL-more | 259 | 71.4 | - | 67.6 | 60.6 |
| Ontonotes | 525 | 82.3 | - | 77.7 | 71.2 |
| WNUT | 103 | 75.7 | - | 57.3 | 53.4 |
| TOTAL | 1107 | 73.6 | - | 69.3 | 61.5 |

Table 36: Accuracy of humans (final redesigned study), context-only biLSTM-CRF, context-ony BERT, and both humans and context-only BERT w.r.t dataset labels.

strength of the selectional preference, the label in the dataset may or may not match the most likely label as determined by the context (human majority). Instead of treating the dataset as a gold standard of context-only labels and determining human or model accuracy, we now analyse the results of the same study treating human labels *and their distribution* as the gold standard and determining how often the dataset label aligns with the most likely label for a given context. With this in mind, we divide the data into three categories based on the strength of the selectional preference as determined by the distribution of the human labels for an instance–

**Strong Preference** There is a majority human label and this is the only label selected by all three annotators. In these instances, there is a strong selectional preference towards the majority label. While other labels may be likely as the human selection may not be exhaustive, their likelihood is much less in comparison. We call these contests as constraining contexts. Their frequency of occurrence is shown in Table 37. 40-50% of examples in the study are constraining contexts. For 90% of the cases, the dataset label matches the most likely label for the context i.e. dataset labels are often in line in with the context-based label when the strength of the preference is strong.

**Weak Preference** There is a majority label but at least one other label is selected by the annotators. These instances have a weaker selectional preference for the majority label. The majority label is more likely than others for the context but others can also be very

| Dataset | # ex | Strong Pref | | Weak Pref | | No Pref |
|---|---|---|---|---|---|---|
| | | % of total | % matches | % of total | % matches | % of total |
| CoNLL-20 | 20 | 20.0 | 100.0 | 80.0 | 75.0 | 0.0 |
| CoNLL-200 | 200 | 41.0 | 67.1 | 45.5 | 54.9 | 13.5 |
| CoNLL-more | 259 | 42.1 | 97.2 | 45.9 | 67.2 | 12.0 |
| Ontonotes | 525 | 60.4 | 92.1 | 36.0 | 73.5 | 3.6 |
| WNUT | 103 | 54.4 | 86.0 | 39.8 | 70.5 | 4.9 |
| TOTAL | 1107 | 51.3 | 88.9 | 41.2 | 61.9 | 7.2 |

Table 37: Percentage of instances with strong, weak and no selectional preferences. Strong preference means that only one (and the same) label was selected by all the annotators. Weak preference means that multiple labels were selected but there was still a majority label. No preference means there was no majority label. For the strong and weak preferences, % matches denote the percentage of instances in the category for which the dataset label matches the human label.

likely. Table 37 shows that these form 35-45% of the contexts in the study. The dataset label aligns for 62% of the cases. The percentage matching labels is lower than that for strong preferences which is expected. The absolute numbers offer insight into how prevalent they are and how much harder it is to recognize them.

**No preference** There is no majority label and two or more labels are equally likely for the context i.e. the context is very ambiguous. In such cases, the word identity is needed to determine the label. Such contexts are uncommon (Table 37).

## 7.4 Automatic Identification of Constraining Contexts

Automatic identification of constraining contexts is necessary to build the proposed hybrid NER model. An ideal NER model would be built keeping in mind the limits posed on generalization due to contexts not always being strongly predictive. If the context is constraining, do not use the word identity and determine the type using context alone. This would allow recognition of all unseen and rare entities. If the context is not constraining, then include word identity to make a prediction. The accuracy, in this case, would be limited by the entities the model has seen.

| Dataset | # ex | Strong Pref | | | Weak Pref | | |
|---|---|---|---|---|---|---|---|
| | | ctx | mlm | ctx/mlm | ctx | mlm | ctx/mlm |
| CoNLL-20 | 20 | 75.0 | 100.0 | 100.0 | 75.0 | 50.0 | 75.0 |
| CoNLL-200 | 200 | 80.5 | 80.5 | 86.6 | 52.8 | 54.9 | 68.1 |
| CoNLL-more | 259 | 88.1 | 78.0 | 93.6 | 63.9 | 59.7 | 79.0 |
| Ontonotes | 525 | 84.9 | 57.7 | 91.2 | 69.3 | 56.6 | 85.2 |
| WNUT | 103 | 84.2 | 49.1 | 86.0 | 48.8 | 31.7 | 53.7 |
| TOTAL | 1107 | 84.7 | 64.3 | 90.5 | 57.3 | 49.7 | 70.1 |

Table 38: Accuracy of ctx-only BERT and MLM-KNN BERT w.r.t. to the majority human label, for contexts with strong and weak preferences.

We explore two methods for the automatic identification of constraining contexts. First, we use context-only BERT. The accuracy of the context-only BERT w.r.t. human majority labels is shown in Table 38 (column *ctx*). For contexts with strong selectional preferences i.e. constraining contexts, the label predicted by context-BERT is the same as the human majority label for 85% of the cases. This number drops to 57% for contexts with weak preferences. There is still a weak preference and therefore the prediction is correct for the majority of the instances. But other labels are very likely too and therefore the accuracy is much lower than that of contexts with strong preferences. Ideally, we would like a model that recognizes only constraining contexts (high recall) and no other contexts (high precision). Unlike full models which are always confident in their prediction, context-only model exhibit a spread of probabilities for their predictions. Unfortunately, these do not align with the strength of the preference. With a higher probability cutoff, we are able to improve precision by recognizing less number of non-constraining contexts. At the time, we also lose recall by identifying less number of constraining contexts too.

We propose another method (*MLM-kNN*) for identifying constraining contexts. Instead of training a model on the target task data which may strongly learn dataset-specific contextual patterns, we directly use pre-training models without any finetuning. Below we describe the method, also shown in Figure 12 –

1. From the training set, collect the words of each type including non-entity.

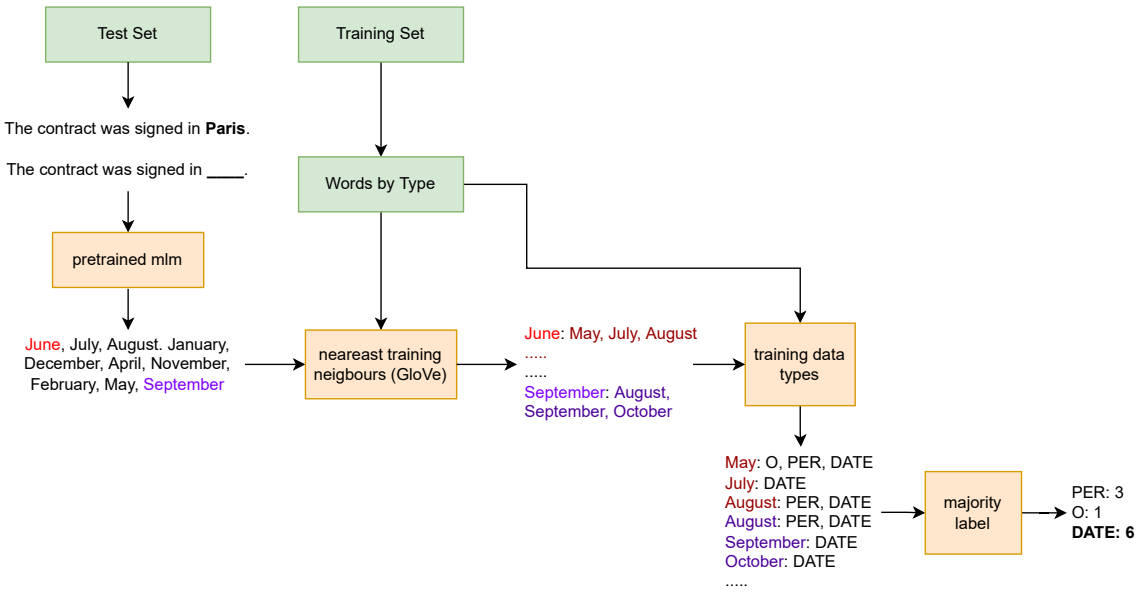



Figure 12: MLM-kNN for determining entity types based solely on the context.

2. For each word in the test set, mask the word and collect the top n predictions using a masked language model (bert-large-cased). The predictions are often simple words/names because they are from the model vocabulary. Word continuations (beginning with ## are discarded).

3. For each predicted word, determine the k nearest neighbours from the set of words in step 1 using GloVe word representations.[23]

4. Collect all the entity types (from step 1) corresponding to each of the nearest neighbour words. If a word appears with multiple types, collect all the relevant tags.

5. The majority tag is considered the final predicted label. If there is a clear majority tag (no ties), we consider the context constraining.

The accuracy of the MLM-kNN w.r.t. the human majority labels is shown in Table 38 (column *mlm*). We use n=10 and k=5. and The method has lower accuracy than context-only BERT. Moreover, it suffers from the same drawbacks as the context-only models. It has

[23]BERT word representations does not work well for this for similarity computation

| Dataset | ctx | mlm |
|---|---|---|
| CoNLL-20 | 85.0 | 60.0 |
| CoNLL-200 | 54.5 | 54.5 |
| CoNLL-more | 67.6 | 59.5 |
| Ontonotes | 77.7 | 48.5 |
| WNUT | 57.3 | 32.0 |
| TOTAL | 69.3 | 50.8 |

Table 39: Percentage of instances where the predicted label matches the dataset label.

| Dataset | dataset | ctx | mlm | ctx/mlm | ctx+mlm |
|---|---|---|---|---|---|
| CoNLL-20 | 100.0 | 95.0 | 90.0 | 100.0 | 85.0 |
| CoNLL-200 | 80.0 | 83.0 | 85.5 | 91.5 | 77.0 |
| CoNLL-more | 93.8 | 90.0 | 82.6 | 95.8 | 76.8 |
| Ontonotes | 93.1 | 86.8 | 69.3 | 93.5 | 62.6 |
| WNUT | 89.3 | 88.3 | 68.0 | 89.3 | 67.0 |
| TOTAL | 90.7 | 87.2 | 75.6 | 93.4 | 69.3 |

Table 40: Percentage of instances where the label (dataset or predicted) matches any label selected by humans, not just the majority label.

a fairly good recall for constraining contexts but others contexts are recognized as well. We perform an oracle experiment to determine if context-only model and MLM-KNN model are recognizing the same set of contexts (col ctx/mlm in Table 38). The oracle has significantly higher accuracy i.e. both models do not always recognize the same contexts.

The recognition of constraining contexts is vital to building a generalization limit-aware NER system. However, the identification of constraining contexts is hard. Both methods are able to identify constraining contexts with good recall, but also identify contexts with weaker preferences. Cutoffs based on confidence level did not help either. An actual weak preference might seem like a strong preference to the model because of the frequency and distribution of similar contexts as well as the types of entities that appear in those contexts in the model data. In addition to the difficulty of recognizing constraining contexts, there are other factors as well that make developing such a NER model hard. Even if we could recognize constraining contexts, they are still preferences at the end of the day. For 10% of the cases for strong preferences, the dataset label did not match the majority label (column % matches under strong pref in Table 37). Therefore, we did not pursue this direction further. Instead, we use both of these to generate plausible labels for a given context (next section).

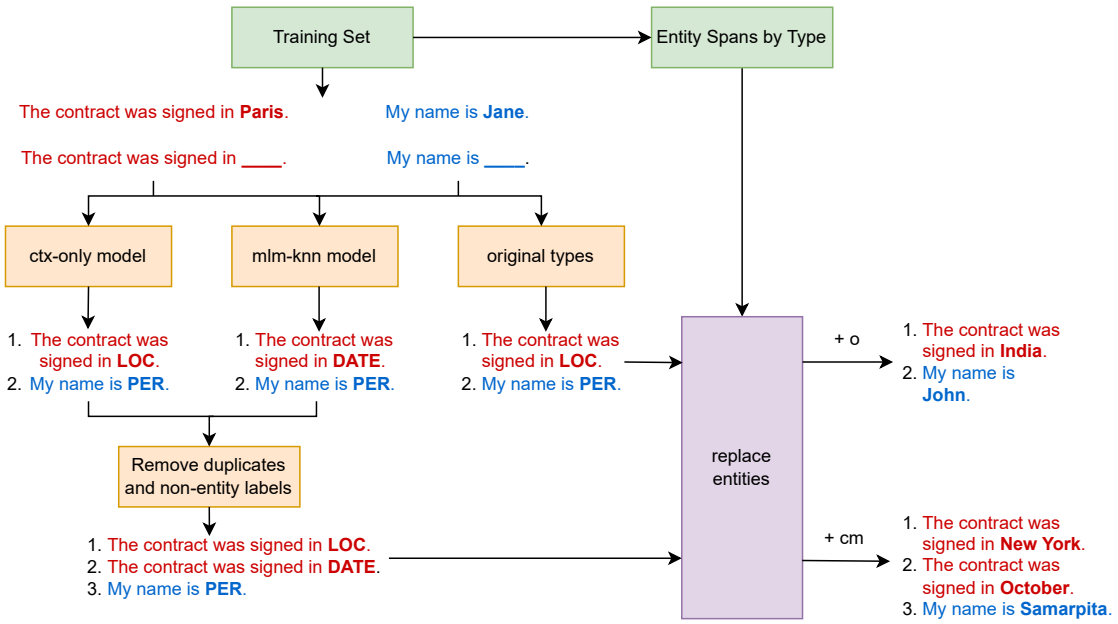



Figure 13: Generating new examples from the training data based on the original types and predicted types wth the ctx-only and MLM-kNN models.

## 7.5 Data Augmentation via Context Label Set Expansion

The labels generated by both context-only models and MLM-KNN do not always match the dataset label (Table 39). The average agreement of the dataset labels with labels predicted by the context-only model and MLM-KNN for the instances in the human study is 69.3% and 50.8% respectively. Despite this, the labels generated by both methods are valid for the context often (Table 40), as determined by the set of labels selected by humans. The label predicted by the context-only model is selected at least once by humans in 87.2% of the cases. That predicted by MLM-kNN is selected in 75.6% instances. Both methods do not always predict the same label. An oracle combination of the two methods yields an accuracy of 93.4%. By switching the entity in the dataset with an entity of the type determined by these methods, we can potentially generate new valid examples for training models. The new entities can be selected from the training data itself, thereby requiring no external resources. The example generation process is illustrated in Figure 13.

### 7.5.1 Models

We build models trained on a combination of the original data and the generated examples, along with several baselines to ensure the effectiveness of the methods. Following are the different models–

**base** The base model (bert-large-cased) is trained on the original training data. With models trained on the full training splits of thousands of sentences, there is often little scope for improvement, especially in-domain. At the same time, developing models that can work well with limited training data is important because the process of collecting human-labeled data is cumbersome and expensive. Therefore, we work in the setting of limited training data. We downsample training sets to 100 sentences. For a similar-sized development set, we downsample it to 100 sentences as well. We draw three different samples and the model is trained on each sample with a different seed.

**+ o** We augment the original training data with entity-switched sentences. Each entity in the training set is switched with another entity of the same type from the set of entities in the training data itself. Therefore no new entities are introduced. New examples are created by new entity+context combinations. The number of sentences added is equal to the number of entities in the training set. This method was introduced in Dai and Adel (2020) for NER in biomedical and material science text.

**+ o + o** The process is repeated twice to see if there are continued gains by creating more such examples. This setup also serves as a control to compare augmentation by new plausible types. The purpose is to ensure that any improvement is not merely due to more training data since this method will always have the most training sentences.

**+ o + c** The base model is trained on the original 100 training examples and context-only predictions are made for the training set entities. The full entity span is masked instead of individual words as before because the replacement for augmentation will

be done at the span level. We are now working with the training set and therefore know the gold labels and the location of the entity spans. For the earlier analysis, we were noting the performance on the test set where prediction is at the word level without the knowledge of whether it is a part of an entity or not. The masked spans are replaced by a training data entity with the same type as the predicted label. If the predicted label is O (non-entity), the instance is discarded.

**+ o + m** MLM-kNN is used to determine context-based types for each entity span in the training set. We still use n=10 for MLM but k is reduced to 3 for kNN because of the limited number of entities of each type in the small training set. The masked spans are replaced by a training data entity with the same type as the predicted label. If the predicted tag by MLM-kNN is the same as the predicted tag via context-only models, the same replacement entity is used. If the predicted label is O (non-entity), the instance is discarded.

**+ o + cm** The dataset, context-only prediction and MLM-kNN labels do not always align. Therefore, we include examples generated by all three methods (o + c + m). If the predicted tag for any entity in 'c' and 'm' matches, it is included only once.

### 7.5.2 Results

We evaluate models on the datasets from Chapters 3 (CoNLL + Entity-Switching), 4 (NYT) & 5 (TTC). We also perform a cross-domain evaluation across the six different genres in Ontonotes (Pradhan and Xue, 2009) which also consists of more entity types. We report the average span-level F1 for the three models trained on the different samples, along with the average final number of training sentences in each case.

Results for the model trained on CoNLL are shown in Table 41. We evaluate it on the original test set, along with the entity-switched Indian and Vietnamese test set where all *PER*, *LOC* and *ORG* entities are replaced. There are 166 entities on average in the training splits. Therefore with the basic permutation using original types, we end with a total of 266

| | #sents | Original | Indian | Vietnamese | Avg Δ |
|---|---|---|---|---|---|
| base | 100 | 75.7 | 57.3 | 45.9 | - |
| + o | 266 | 76.8 | 61.2 | 48.5 | 2.50 |
| + o + o | 432 | 75.2 | 57.6 | 45.9 | -0.07 |
| + o + c | 306 | 76.6 | 60.9 | 48.6 | 2.39 |
| + o + m | 339 | **78.4** | **61.5** | **50.5** | **3.81** |
| + o + cm | 350 | 77.8 | 60.7 | 49.1 | 2.84 |

Table 41: Span F1 of the model trained on CoNLL-100 when evaluated on the original test set, and entity-switched Indian and Vietnamese test sets. The number of training sentences is averaged over the three samples. Average Δ is the change in F1 over the base model averaged over the test sets.

| | #sents | 2015 | 2016 | 2017 | 2018 | 2019 | Avg Δ |
|---|---|---|---|---|---|---|---|
| base | 100 | 32.1 | 31.9 | 31.9 | 31.2 | 36.8 | - |
| + o | 161 | 39.4 | 40.9 | 40.0 | 38.8 | 45.3 | 8.10 |
| + o + o | 223 | 41.4 | 42.0 | 41.0 | 40.0 | 46.1 | 9.33 |
| + o + c | 165 | 39.6 | 41.2 | 40.8 | 39.1 | 46.3 | 8.62 |
| + o + m | 174 | **41.6** | **42.3** | **42.2** | **40.3** | **47.8** | **10.06** |
| + o + cm | 178 | 41.4 | 41.6 | 41.3 | 39.5 | 47.0 | 9.38 |

Table 42: Span F1 of the model trained on the downsampled 2014 split of TTC when evaluated on each of future test splits (2015-2019). The number of training sentences is averaged over the three samples. Average Δ is the change in F1 over the base model averaged over the test sets.

sentences. Even with the basic permutation (+ o), we see an average improvement of 2.50 points F1 across the three test sets. Repeating another round of the basic permutation (+ o + o) does not help. In fact, it leads to a very slight drop in F1. Augmentation with entities using labels from the context-only models (+ o + m) does not yield a better model either, though it is still better than the base model. This can be attributed to a certain extent to the low recall of a context-only model trained on just 100 sentences. Out of 166 entities, only 40 were predicted as an entity. Augmentation with the labels from MLM-kNN ( + o + m) yields the best model with an average improvement of 3.81 points F1. A combination of the two (+ o + cm) results in an improvement over context-only version but not over the MLM-kNN version. The best-performing model is not the one with the most amount of training data, confirming the effectiveness of the methods.

Similar to CoNLL, we evaluate models trained on the 2014 split of TTC on all the

| | | a | b | c | e | f | m | n | s | o | Avg Δ |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | base | 78.6 | 84.6 | 91.1 | 84.9 | 87.3 | 82.3 | 86.3 | 86.1 | 80.9 | - |
| | + o | **80.4** | 84.0 | 90.9 | **87.3** | 88.2 | **84.4** | 86.8 | 86.3 | 81.5 | 0.86 |
| InD | + o + o | 80.2 | 82.8 | **91.5** | 83.8 | **88.8** | 84.2 | 87.5 | 86.1 | **83.0** | 0.67 |
| F1 | + o + c | 79.9 | 84.0 | 89.6 | 85.9 | 88.0 | 83.9 | **87.9** | 86.0 | 81.9 | 0.57 |
| | + o + m | 80.2 | 85.0 | 90.9 | 85.6 | 87.5 | 84.0 | 86.7 | 86.5 | 81.9 | 0.70 |
| | + o + cm | 79.8 | **85.5** | 90.6 | 86.4 | 87.2 | **84.4** | 86.6 | **86.9** | 82.9 | **0.91** |
| | base | 73.7 | 75.8 | 64.3 | 77.5 | 72.9 | 70.6 | 77.0 | 59.7 | 78.6 | - |
| | + o | 75.7 | 75.4 | 64.5 | **81.0** | 74.4 | 76.4 | 77.5 | 61.4 | 78.1 | 1.60 |
| OOD | + o + o | **76.4** | 75.7 | 65.2 | 77.6 | 75.9 | **76.8** | 78.2 | 61.6 | **80.4** | 1.97 |
| F1 | + o + c | 75.3 | 75.6 | **65.6** | 79.7 | 75.3 | 75.1 | 78.4 | 62.2 | 79.6 | 1.88 |
| | + o + m | 76.0 | 78.9 | 64.2 | 80.6 | 74.8 | 76.4 | 78.7 | 64.2 | 79.6 | 2.60 |
| | + o + cm | **76.4** | **80.3** | 62.8 | 79.5 | **77.1** | 76.6 | **79.2** | **65.9** | 80.0 | **3.09** |
| | base | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | |
| | + o | 169 | 176 | 228 | 175 | 188 | 182 | 185 | 196 | 175 | |
| # | + o + o | 338 | 253 | 355 | 251 | 275 | 265 | 270 | 292 | 249 | |
| sents | + o + c | 193 | 186 | 276 | 191 | 207 | 215 | 218 | 228 | 194 | |
| | + o + m | 205 | 219 | 280 | 212 | 243 | 231 | 237 | 246 | 209 | |
| | + o + cm | 207 | 222 | 305 | 214 | 245 | 233 | 241 | 256 | 212 | |

Table 43: Span F1 of the models trained on each of the topics when evaluated on the same topic (InD) and the remaining eight topics (OOD). For OOD, the average over the remaining eight topics is reported. The number of training sentences is averaged over the three samples. Average Δ is the change in F1 over the base model averaged over the nine models trained on different topics.

future splits (2015-2019). Results are reported in Table 42. The performance trends are similar to that of CoNLL, except that two rounds (+ o + o) of basic permutation show an improvement over one round (+ o). Further augmentation using MLM-kNN based labels is still the best model.

We furthermore evaluate models trained on each topic in NYT. The first set of rows in Table 43 show the F1 of models trained on each of the topics when evaluated on the same topic. The second set of rows shows the average F1 when evaluated on each of the remaining eight topics. The best-performing model varies by topic. However, when averaged over the nine model topics, the most improvement comes from the combined data augmentation (+ o + cm) for both in-topic and out-of-topic testing. Notably, the improvement is higher for out-of-topic testing, even though there are no new entities.

| | | nw | bn | bc | tc | mz | wb | Avg Δ (4) | Avg Δ (5) |
|---|---|---|---|---|---|---|---|---|---|
| | base | 68.5 | 71.3 | 47.5 | 0.0 | 67.4 | 16.6 | - | - |
| | + o | 71.8 | 71.7 | 50.4 | - | 68.4 | 18.0 | 1.90 | 1.81 |
| InD | + o + o | 71.6 | 71.9 | 50.7 | - | **69.4** | **20.2** | 2.20 | **2.49** |
| F1 | + o + c | 71.8 | **72.9** | 50.0 | - | 66.3 | - | 1.56 | - |
| | + o + m | **72.6** | 71.6 | 51.2 | - | 66.8 | 19.5 | 1.85 | 2.07 |
| | + o + cm | 71.4 | 72.0 | **53.1** | - | 69.1 | - | **2.71** | - |
| | base | 53.9 | 54.9 | 42.8 | 0.0 | 44.2 | 7.3 | - | - |
| | + o | 56.5 | 56.2 | 45.8 | - | 47.7 | 6.7 | 2.58 | 1.93 |
| OOD | + o + o | 57.0 | 56.5 | 47.1 | - | 48.0 | **15.7** | 3.18 | **4.22** |
| F1 | + o + c | **57.3** | **57.2** | 46.1 | - | 48.0 | - | 3.18 | - |
| | + o + m | 57.2 | 55.7 | 46.1 | - | 48.8 | 9.0 | 2.99 | 2.72 |
| | + o + cm | 56.2 | 55.0 | **47.6** | - | **51.5** | - | **3.62** | - |
| | base | 100 | 100 | 100 | 100 | 100 | 100 | | |
| | + o | 317 | 288 | 186 | - | 263 | 136 | | |
| # | + o + o | 534 | 475 | 272 | - | 427 | 173 | | |
| sents | + o + c | 372 | 334 | 195 | - | 279 | - | | |
| | + o + m | 421 | 372 | 212 | - | 335 | 142 | | |
| | + o + cm | 441 | 386 | 216 | - | 340 | - | | |

Table 44: Span F1 of the models trained on each of the genres when evaluated on the same genre (InD) and the remaining five genres (OOD). For OOD, the average over the remaining five genres is reported. The number of training sentences is averaged over the three samples. Average Δ (4) is the change in F1 over the base model averaged over the four models trained on different genres (nw, bn, bc, mz). Average Δ (5) also includes wb.

Lastly, we evaluate models trained on each genre in Ontonotes. Unlike the above datasets, Ontonotes also has a more diverse set of entity labels (Person, Location, Organization, Geopolitical-entity, Facility, Product, Date, Time, Ordinal, Cardinal, Law, Money, Work of Art, Quantity, Event, Percent, Language, Nationality/Religion/Political Party). Some of the genres (tc and wb) are very sparse in entities. With telephone conversations, the F1 was 0 even with the base model and we exclude it from the average reporting. For web, no span was recognized as an entity, there we exclude it from the average reporting too. Results are shown in Table 44. The results are similar to that of NYT, with the combination (+ o + cm) model resulting in the best performance overall. The change over the model is higher for OOD evaluation.

In summary, data augmentation via entity permutation in the training set is an effective tool for improving model performance in low-resource settings. Even without introducing

new entities, such a permutation results in a significantly better model. However, repeating the process does not always lead to continued gains. A better model can instead be obtained by a smarter augmentation strategy based on the set of plausible labels for a context. The label set for a context can be expanded using context-only models or MLM-kNN. The best-performing model varies between $+ o + m$ and $+ o + cm$ across datasets.

## 7.6 Conclusion

We performed a human study with the task of determining the entity type solely from the context, without the knowledge of the word itself. The goal was to answer two questions—

1. **Feasibility of More Context Utilization**: Can humans identify entity types correctly when the model is incorrect? The answer to the question is yes. However, the task is incredibly hard even for humans and there is only limited scope for better context utilization.

2. **Feasibility of Model for Better and Realistic Generalization**: Are contexts usually constraining and ambiguous? We found that roughly half the contexts are constraining. Despite this, it is challenging to build a generalization limit-aware model due to the difficulty in the automatic identification of constraining contexts. However, the developed methods can be used to expand the set of labels plausible in a context to perform training data augmentation via entity-switching.

# CHAPTER 8

# Fine-grained NER Challenges

The work in this thesis and much of NER literature focuses on coarse-grained entity types, namely person, location, organization and miscellaneous entities. While we did perform experiments with the 18 entity types in Ontonotes wherever possible, most of the analysis in this thesis is hard to perform even for coarse-grained entity types and its extension to fine-grained types would require significant effort. Developing entity-switched datasets needed manual human intervention for organizations for a coherent replacement. Such an intervention would be difficult for a large number of entity types if necessary. The datasets to study domain and temporal effects involved collecting a sufficient number of examples in each of the sub-domains and time periods. With a large number of types, even larger datasets would be necessary to have a sufficient number of examples of each entity type for a reliable study. The context-based human evaluation was so hard that we had to drop even the miscellaneous entity types to make the task easier for the human raters.

Nonetheless, there is a vast amount of work on the recognition and typing of diverse fine-grained entity types that we discuss here for completeness. We focus our discussion on broad entity types that can be found in general text such as news and social media. We do not discuss work on the recognition of entity types specific to specialized domains that would involve domain adaptation. Most of the work in this area focuses on entity typing (FNET), a subtask of named entity recognition (FNER). We discuss such work as well since the core challenges remain the same for both and named entity recognition can always be performed in two steps of span detection and typing. We also discuss relevant work on coarse-grained entity recognition that we believe has the potential for fine-grained entity recognition. Lastly, we present some of our own work with initial experiments in this direction. The chapter is divided into four sections, discussing four main questions–

1. What should be the target set of entity types?

2. How can we develop training data for such a large number of entity types?

3. How do we evaluate models for such a large number of entity types?

4. How can we develop models that can handle new entity types without retraining?

## 8.1 Entity Types

We discuss the entity types in popular datasets, various proposed entity ontologies, and the entity types available in off-the-shelf models.

### 8.1.1 Datasets and Ontologies

CoNLL '03 (Tjong Kim Sang and De Meulder, 2003), Ontonotes (Pradhan and Xue, 2009), WNUT '16 (Strauss et al., 2016) and WNUT '17 (Derczynski et al., 2017) are undoubtedly few of the most popular named entity recognition datasets. While CoNLL has only four coarse-grained entity types, the other three have a more diverse set of entity types at a finer granularity. Specifically, Ontonotes has 18 entity types, and WNUT '16 and '17 have 10 and 6 entity types respectively. MultiNERD (Tedeschi and Navigli, 2022), a very recent dataset, has 15 types of entities. The set of tables in each dataset are shown in Table 45.

Besides datasets developed with specific entity types, there have also been various entity ontologies proposed for a much larger set of entity types. Sekine et al. (2002) designed an entity type hierarchy with 150 types by manually assigning tags to entities extracted from various sources. Yosef et al. (2012) derived an entity type hierarchy with 505 types from the YAGO Knowledge Base (Suchanek et al., 2007). The large number of types include qualitative labels such as person/good person and event/happening/beginning. Ling and Weld (2012) created FIGER which consists of 112 types derived from Freebase (Bollacker et al., 2008). Murty et al. (2018) created TypeNet with about 1900 types by manually aligning Freebase types with the WordNet (Miller, 1995) hierarchy.

| Dataset | Entity Types |
|---|---|
| CoNLL '03 | Person, Location, Organization, Miscellaneous |
| Ontonotes | Person, Location, Geo-political Entity, Facility, Organization, Nationality/Religion/Political Party, Product, Work of Art, Event, Date, Time, Law, Language, Cardinal, Ordinal, Quantity, Percent, Money |
| WNUT '16 | Person, Location, Company, Sports Team, Facility, Product, Music Artist, Movie, TV Show, Other |
| WNUT '17 | Person, Location, Corporation, Group, Product, Creative Work |
| MultiNERD | Person, Location, Organization, Animal, Biological entity, Celestial Body, Disease, Event, Food, Instrument, Media, Plant, Mythological entity, Time, Vehicle |

Table 45: Entity Types in popular NER datasets.

Fine-grained entity typing poses a challenge as to whether the entity type should depend on the context or the surface form. With coarse-grained entity recognition without overlapping types, there is only one correct label based on the joint resolution of entity and context, though it could be inferred from either one as well for many cases. However, for fine-grained entity recognition where types belong to a hierarchy, the most likely type based on the surface form alone and that based on the context can differ. For example, in the sentence, "Bill Gates donated to several organizations", Bill Gates should be a philanthropist based on the context but he is also a businessman in real life. While some prior work defaulted to surface form for simplicity, Gillick et al. (2014) argue that the label should depend on context. With that in mind, they create an ontology of 87 types and also manually annotate 12k entities Ontonotes for context-dependent fine-grained types. To develop the training data, they use heuristics to ensure the distantly supervised types are based on the context.

Choi et al. (2018) take a different view, setting up the task to predict a set of free-form phrases for the entity since they found that annotators of Ontonotes in Gillick et al. (2014) could not find a suitable type in the ontology for half of the mentions. They create a dataset of 6000 entities with 2500 types.

There seems to be no consensus on the hierarchical organization scheme with different works using different ontologies. However, FIGER is the most widely one in literature.

| Model | Entity Type Set | URL |
|---|---|---|
| NLTK | PER, ORG, GPE | Link |
| spaCy | Ontonotes | Link |
| Stanza | CoNLL, Ontonotes | Link |
| AllenNLP | CoNLL, Ontonotes | Link |
| Flair | CoNLL, Ontonotes | Link |
| Polyglot | PER, ORG, LOC | Link |

Table 46: Entity Types in public off-the-shelf models.

| Model | Entity Type Set | URL |
|---|---|---|
| Google Cloud | Per, Org, Loc, Event, Work of Art, Consumer Good, Phone Number, Address, Date, Price, Other | Link |
| Amazon Comprehend | Per, Org, Loc, Event, Commercial Item, Date, Quantity, Title, Other | Link |
| Microsoft Azure | Person, Person Type, Loc (GPE, Structural, Geographical), Org (Medical, Stock Exchange, Sports), Event (Cultural, Natural, Sports), Product, Skill, Address, Phone Number, Email, URL, IP, DateTime (Date, Time, Date Range, Time Range, Duration, Set), Quantity (Number, Percent, Ordinal, Age, Currency, Dimensions, Temperature) | Link |
| IBM Cloud | See link | Link |

Table 47: Entity Types in cloud computing APIs.

### 8.1.2 Off-the-shelf Models

While there are plenty of datasets on fine-grained entity recognition, public downloadable models are limited to the 4 types in CoNLL and the 18 types in Ontonotes. We report the entity types in these models in Table 46.

Paid APIs available in various cloud computing platforms have a larger set of entity types. These may reflect the entity types that are relevant to their customers. Microsoft Azure and IBM Cloud have a huge entity of fine-grained entity types, though their accuracy is unknown. We report the entity types available through these APIs in Table 47.

## 8.2 Training Data

Manually annotating sufficient training data for each type in the entity hierarchy would be laborious and possibly even infeasible. We discuss automatic methods for obtaining

training data for fine-grained entity recognition, namely distant supervision, self-labeling and synthetic data generation.

### 8.2.1 Distant Supervision

Wikipedia with its link structure is the most common source of training data for named entity recognition, both coarse-grained and fine-grained (Kazama and Torisawa, 2007b; Nothman et al., 2008; Yosef et al., 2012; Ling and Weld, 2012; Al-Rfou et al., 2015; Ghaddar and Langlais, 2017, 2018a; Abhishek et al., 2019).

Kazama and Torisawa (2007b) work with the CoNLL entity types, retrieving entity types from the first sentence of Wikipedia articles. Nothman et al. (2008) classify articles into person, location and organization by bootstrapping with a set of hand-labeled seed articles. Al-Rfou et al. (2015) use the common type but generate data in 40 languages. Furthermore, they improve the recall of entities by identifying unlinked entities via phrase matching. On the other hand, Ghaddar and Langlais (2017) improve the entity recall by adding coreference mentions of linked spans.

Yosef et al. (2012); Ling and Weld (2012); Ghaddar and Langlais (2018a) work with fine-grained entity types and use entity types from knowledge bases to annotate linked (and sometimes unlinked) spans in Wikipedia. Abhishek et al. (2019) develop a multi-stage pipeline to label Wikipedia, creating a corpus of 32M sentences. They improve both the recall and the precision of the entities through the pipeline. In the first stage, they improve precision by classifying linked spans into entity and non-entities since all links may not refer to an entity in the hierarchy. They then improve recall using several heuristics and by only selecting sentences where unmarked tokens are most likely to be non-entity words based on their identity.

### 8.2.2 Self-labeling

Self-labeling involves training a model on pseudo-labels generated by another model for the same task. The goal is to expand the amount of training data thereby improving the coverage of entities even if the labels are noisy. The label-generating model is generally trained on gold standard data as we did for temporal model adaptation. However, it could also be trained without in-domain gold standard labels. BOND (Liang et al., 2020) labels in-domain dataset-specific text with knowledge bases and gazetteers to generate training data. GeNER (Kim et al., 2022) labels target-label relevant text from Wikipedia retrieved with a QA retriever instead. Both methods then train BERT on this data. It is trained further for several iterations on soft pseudo labels produced by the model in the previous iteration. Both methods use dataset-specific labeled development sets for early stopping. Evaluation is performed on a variety of datasets with entity types including person, location, organization, product, diseases, chemical, movies etc. While self-labeling has only been performed for data-specific labels in prior work to the best of our knowledge, it could be potentially used for the fine-grained distantly supervised datasets described in the last section.

### 8.2.3 Synthetic Data Generation

Lastly, we discuss a fairly new avenue for creating training data. Synthetic data can be generated using language models and then labeled via distant supervision or self-labeling.

He et al. (2022) propose a framework called "generate, annotate, and learn (GAL)" to generate task-specific unlabeled data. They generate synthetic sentences with language models either fine-tuned on task-specific unlabeled data or prompted with a few examples. The generated data is labeled with pseudo labels using the best available classifier and a new model is trained on a combination of labeled and self-labeled data. While NER was not one of their target tasks, it could benefit from this method.

Synthetic data can also be generated from knowledge graphs. In our work (Agarwal et al., 2021a), we build a multi-stage pipeline and convert the full Wikidata KG into synthetic

| KG triples | Generated Sentence |
|---|---|
| Michelle Obama height +71 inch. | Michelle Obama is 71 inches tall. |
| Spork EP performer The Shins, instance of Extended play, publication date 01 January 1995, genre Indie rock, language of work or name English language. | Spork EP is the English language extended play by the band The Shins. It was released in 1995. Spork EP is an Indie rock genre. |
| Michelle Obama position held First Lady of the United States, First Lady of the United States replaced by Melania Trump, First Lady of the United States start time 20 January 2009, First Lady of the United States replaces Laura Bush, First Lady of the United States end time 20 January 2017. | Michelle Obama served as the First Lady of the United States from 2009 to 2017, replacing Laura Bush. She was succeeded by Melania Trump. |
| 10x10 Photobooks inception 00 2012, instance of Nonprofit organization. | 10x10 Photobooks, founded in 2012 is a nonprofit organization. |

Table 48: Examples of verbalization (inference) from KELM (Agarwal et al., 2021a)

natural language sentences. Since knowledge graphs represent entities and their relations, the generated sentences are rich in diverse entities and potentially a good source of training data for NER. Some examples from this dataset are shown in Table 48.

Ding et al. (2020) approached synthetic data generation differently by generating labeled data instead of unlabeled text. They linearize training data sentences by adding entity labels as span markers in the sentence. They train a language model on this data and use it to generate synthetic data for training data augmentation. The generated data consists of entity marker spans and is therefore labeled. Furthermore, they even utilize unlabeled and noisy KB-labeled text for language model training by prepending condition tags [labeled],[unlabeled],[KB] at the beginning of each sentence. Inference is performed with the [labeled] tag. They evaluate their methods on a variety of datasets and observe huge improvements with low-resource settings. Their method can potentially be used to generate sentences with fine-grained entity types by mixing existing gold-standard datasets and distantly supervised KB-labeled datasets.

Lastly, we propose the use of language models to create synthetic gazetteers instead of mining them from the web. These gazetteers can then be used to label unlabeled text or to

| *Generate a list of Indian names* for boys | *Generate a list of Vietnamese names* | *Generate a list of dates* |
|---|---|---|
| 1. Aarav | 1. Nguyen | 1. January 1st |
| 2. Anil | 2. Trinh | 2. February 2nd |
| 3. Amit | 3. Pham | 3. March 3rd |
| 4. Arjun | 4. Hoang | 4. April 4th |
| 5. Arnav | 5. Le | 5. May 5th |
| 6. Dev | 6. Vu | 6. June 6th |
| 7. Dhruv | 7. Do | 7. July 7th |
| 8. Gopal | 8. Dang | 8. August 8th |
| 9. Hari | 9. Bui | 9. September 9th |
| 10. Inder | 10. Nguyen | 10. October 10th |
| 11. Jay | 11. Vu | 11. November 11th |
| 12. Krish | 12. Trinh | 12. December 12th |

Table 49: Prompting GPT-3 to generate entity lists. The prompt is italicized.

create entity-switched datasets. Some examples of lists generated with GPT-3 text-davinci-003 (Brown et al., 2020) are shown in Table 49. We generated short lists by putting limits on the target length, therefore the coverage that can be achieved with such lists remains to be explored.

## 8.3 Evaluation

Creating evaluation datasets of fine-grained named entity recognition has the same challenge as creating training data. We need sufficient examples for each type for reliable evaluation but unlike the training data, the labels can not be noisy.

Ling and Weld (2012) developed a gold standard human-labeled test dataset corresponding to their developed entity hierarchy types. This is the most commonly used dataset for evaluation. However, Abhishek et al. (2019) noted that while FIGER has 112 types in the hierarchy, only 42 types are present in the test set and 80% of the entities fall under the person, location and organization coarse types despite a much more diverse set of labels in the ontology. Therefore, they also developed a dataset called 1k-WFB-g ensuring sufficient coverage of the entity types.

Fine-grained entity models are also typically evaluated on several NER datasets by

manually mapping the labels. However, such a manual mapping may not be possible for other larger ontologies. Therefore, we explored the possibility of using automatic label mapping between datasets using label embeddings. Unfortunately, this results in labels being matched by theme, rather than type. For example, judge was mapped to law instead of person.

## 8.4 New Unseen Labels

The work discussed till now assumes a fixed set of entity types. However, entity types may evolve or our target entities of interest may change. We now discuss work that can handle new entity types either without any new examples (zero-shot) or with a small set of new examples (few-shot).

### 8.4.1 Label embeddings

Since NER labels are words themselves, learning a representation of labels and word in the same space can allow us to generalize to new labels without retraining models. The label most similar to the entity representation would be selected. Such methods assume that the entity spans are known. Yogatama et al. (2015) project word and label embeddings into same size vectors, maximizing their dot product. However, their input label representations are fixed length one hot vectors of the same size as the label set and therefore the method cannot identify unseen entity types. Ma et al. (2016) tackle this by using word2vec representations (Mikolov et al., 2013) of labels instead of one hot vectors. Furthermore, they train the model on a small set of manually selected prorotypical examples of these label to reduce noise. While their model can work in zero-shot settings, they always assume that there are example available foe each type. Yuan and Downey (2018) develop a similar method but use GloVe word embeddings (Pennington et al., 2014a).

Ghaddar and Langlais (2018b) use a different method for learning representation of word and labels in the same space. They replace words with their fine-grained types in the

sentence, concatenate both versions of the sentence and learn uncased embeddings on it. However, they do not use the method for NER via computing word–label similarity. Instead their goal is to learn a fixed-length gazetteer vector by computing similarity between words and fine-grained labels. The gazetteer vector is then concatenated to the input of typical NER model for CoNLL and Ontonotes.

Recently, Ma et al. (2022) developed a model for NER by learning label embeddings. To the best of our knowledge, this is the only method that uses label embeddings for entity recognition, not just entity typing. The method trains a BERT/GloVe based dual encoder where one encoder encodes each word in a sentence and the other encodes all the labels. The label encoder takes as input the natural language form of the label prepended by its location within the span (begin, inside). The similarity between the two representations is maximized as with prior methods. A two stage finetuning is performed. The model is first trained on the full Ontonotes data, followed by k-shot training on the target dataset. The authors find that the second stage of training is essential to good performance. While they do not perform evaluation on fine-grained entity recognition, this method has potential to work well on it.

### 8.4.2 Nearest neighbouring words

Instead of learning label embeddings directly, another method finds the nearest neighbours with known types of the target entity span. This method is similar to step 3 of MLM-kNN in Chapter 7. Yang and Katiyar (2020) use a biLSTM-CRF or BERT trained on the source NER dataset to generate contextual word representations. These representations are used to find the nearest neighbours with known labels of the target span. The set of words with known labels simply includes k examples for every label. With Ontonotes as the source dataset, the method performed well on multiple target datasets. Similar to the above methods, evaluation is not performed for fine-grained entity recognition, but the method has the potential to work well on it.

### 8.4.3 Prompting

Prompting generative models such as GPT-3 (Brown et al., 2020), T5 (Raffel et al., 2022), BART (Lewis et al., 2020) is another technique used to recognize new labels perform zero-shot and k-shot settings. Cui et al. (2021) train BART with the sentence as input and templates of the form " is <label>" as the output. However, the method is computationally complex since inference involves enumerating all spans and labels to find the spans and corresponding highest-scoring label. Li et al. (2020) therefore propose a template-free method to use such models. They frame NER as a question-answering task. The input to the model consistent of the sentence as context concatenated with the question "what is <label>". The output is then the span corresponding to the label. Liu et al. (2022) extend this method to zer-shot and few-shot setting by training the model first on a QA dataset and then on a few examples of NER.

# CHAPTER 9

# Conclusion

Named Entity Recognition models work impressively well and are used for several practical applications. Therefore, the goal of this thesis was two-fold – 1) Evaluate the robustness of practical NER in-domain models, and 2) Determine the feasibility of developing NER models that can identify contexts with strong predictive clues and hence generalize better. Here, we summarize the work for each.

## 9.1 Practical Model Robustness

While NER models achieve high accuracy on benchmark datasets, their performance suffers in several aspects, either not captured by datasets or underrepresented in them.

In Chapter 3, we study the robustness of NER models in recognizing names from different countries of origin. We develop entity-switched datasets by replacing entities in existing datasets with those from several other countries. We find that names from certain countries are recognized with much higher accuracy than others.

In Chapter 4, we study the robustness of NER models across topics within a seemingly homogenous domain. We collect a new NER dataset of news articles from the New York Times, stratified by news topics. Even within such a dataset, model performance on topics not seen in the training data is significantly lower.

In chapter 5, we study the impact of temporal changes on the model performance on NER and other tasks. We establish terminology and design to study temporal effects. We find that the performance of pre-trained models may or may not deteriorate over time; it is task-dependent. However, better performance can be obtained by retraining on recent data.

## 9.2 Realistic Context-based Generalization

For generalization to unseen names, models need to be able to recognize strong contextual clues. In Chapter 6, we quantify the degree to which models learn words or contexts to make predictions. While high performance can be achieved based on just the word, the same is not true based on just the context.

Not all contextual clues may be strong and therefore the prediction would need to depend on the word identity in such cases. Therefore, in Chapter 7, we conduct human studies to determine how often do context impose strong selectional preferences i.e. are contraining. We find roughly half the contexts to be constraining, and not all of them are recognized by models. However, their automatic identification is hard making the development of a model that explicitly recognizes constraining contexts challenging.

Through the human study in Chapter 7, we recognize that several different entity labels are possible in the same context, and the methods developed to recognize constraining contexts generate entity types beyond the dataset label. Therefore, we use these to expand the set of labels possible for a context and perform entity-switching with these new types to create additional training examples. Entities are drawn from the training data itself and training data augmentation with these new examples helps in low-resource settings.

## 9.3 Future Work

There are several future work directions discussed in each of the chapters that are specifically relevant to each. Here, we discuss future directions that are relevant to the thesis as a whole and important for broader research.

**Specialized Domains** The work done in this thesis used mostly text from news and social media (Twitter). We worked with product reviews for temporal changes but not NER. We also worked with the six genres of Ontonotes (newswire, broadcast news, broadcast

conversation, telephone conversations, web, magazines). However, much more specialized domains such as biomedical text, scientific articles and publications (physics, computer science etc), legal text, financial text etc. remain to be explored.

**Finer-grained and Specialized Entity Types** We worked mostly with common coarse-grained entity types (person, location, organization). Conducting human evaluation was hard even with such a small set of labels. Miscellaneous, the catch-all for other entity types, had to removed in iterations of the human study. For automatic evaluation, we did work with Ontonotes and its larger label set. The role of context could be greater in fine-grained entity classification, especially if the coarse grained label is known though there may be influence by the word due to the presence of prominent figure names. The FIGER hierarchy (**?**) consists of 112 entity types. Anything can be a named entity, such as item of clothing, research technique, disease, gene. Identifying these might seem harder but given a dictionary of words or a domain-relevant pretrained models, it might be simpler since these entity types might be less ambiguous.

**Multiple Languages** The work in this thesis was only on English. How it will translate to other languages remains to be explored. Capitalization is a strong indicator for recognizing named entities in English. For languages such as Hindi without capitalization, will the word itself be as strong of an entity indicator? For languages like Chinese without explicit word segmentation, how difficult will it be to differentiate between entity and its context, and conduct such studies? Such analysis can be pursued as future work.

**Impact of Pre-training Data** While the training data has a bigger impact on model performance, it is essential to study the pre-training data for reliable results. For the study on temporal effects, the knowledge of the pre-training data time period is essential for a clean study. Similarly to study the effect of domains, the knowledge of domains within the pre-training data is essential. Considering the size of the pre-training data and the numerous sources, we did not approach such analyses in this thesis.

**Automatic Domain Identification** Zero-shot fine-grained domain classification is hard (Chapter 4) but would be useful for both training and pre-training data selection. There are also challenges to building such a model such as *i)* the perceived sentence-level domain might not be the same the domain of its source documents, and *ii)* there are several aspects of domain (genre, discourse structure, source). Determining the appropriate design for this task remains future work.

**Temporal Model Deterioration and Re-training** In this thesis, we found that not only tasks experience model deterioration with pre-trained models. Performance may even increase as tasks sometimes become easier with time. With increased character limit on Twitter in 2017, NER became easier, likely due to more context availability. Product reviews became shorter over time, thereby making identifying the sentiment expressed in them easier. Therefore, identifying whether our task of interest has experienced model deterioration is necessary to determine the importance of re-training models. Furthermore, determining the frequency of re-training is needed to be cost-effective since the degree of improvement depends vastly based on the target task as well as time period of interest.

## APPENDIX A: Fine-tuning and Hyperparameters

The biLSTM-CRF models are trained using the code from Lample et al. (2016), modified to add ELMo, to build context-only models, and to perform sentence classification. Hyperparameters for these models can be found in the respective chapters.

BERT and RoBERTa are fine-tuned using the implementation in HuggingFace (Wolf et al., 2020). Hyperparameters are optimized via grid search over the learning rate (3e-05, 5e-06, 5e-06), batch size (2, 4, 8, 16, 32, 64) and number of epochs (1, 2, 3, 4, 5, 6) for models trained on the full training sets. The same learning rate is used for the word-only models, but the batch size and the number of epochs are optimized. The context-only model is the full model with modified inference and therefore requires no separate hyperparameter tuning. The best checkpoint of the final model on the development set is selected. Following are the hyperparameters for each dataset–

**CoNLL** Hyperparameters details are below. The model is bert-large-cased.

| | Full model | | | Word-only model | | |
|---|---|---|---|---|---|---|
| | LR | BS | EP | LR | BS | EP |
| CoNLL | 3e-05 | 16 | 4 | 3e-05 | 64 | 1 |

Table 50: Hyperparameters, namely the learning rate (LR), the total batch size (BS) and the number of epochs (NE) for the full and word-only models trained on CoNLL '03.

**Ontonotes** Hyperparameters are optimized for each genre. The model is bert-large-cased.

| | Full model | | | Word-only model | | |
|---|---|---|---|---|---|---|
| | LR | BS | EP | LR | BS | EP |
| nw | 3e-05 | 16 | 3 | 3e-05 | 64 | 3 |
| bn | 5e-05 | 16 | 5 | 5e-05 | 64 | 3 |
| bc | 3e-05 | 8 | 6 | 3e-05 | 64 | 5 |
| tc | 3e-05 | 8 | 2 | 3e-05 | 64 | 3 |
| mz | 5e-05 | 8 | 5 | 3e-05 | 64 | 1 |
| wb | 3e-05 | 8 | 5 | 5e-05 | 64 | 2 |

Table 51: Hyperparameters, namely the learning rate (LR), the total batch size (BS) and the number of epochs (NE) for the full and word-only models trained on Ontonotes.

**NYT** Hyperparameters are optimized for each sub-domain. The model is bert-large-cased.

| | Full model | | | Word-only model | | |
|---|---|---|---|---|---|---|
| | LR | BS | EP | LR | BS | EP |
| arts | 3e-05 | 16 | 4 | 3e-05 | 16 | 4 |
| business | 5e-05 | 8 | 2 | 5e-05 | 8 | 2 |
| classified | 5e-05 | 4 | 1 | 5e-05 | 4 | 1 |
| editorial | 5e-05 | 8 | 2 | 5e-05 | 8 | 2 |
| foreign | 5e-05 | 4 | 2 | 5e-05 | 4 | 2 |
| metropolitan | 5e-05 | 16 | 2 | 5e-05 | 16 | 2 |
| national | 5e-05 | 16 | 2 | 5e-05 | 16 | 2 |
| sports | 3e-05 | 8 | 3 | 3e-05 | 8 | 3 |
| others | 5e-05 | 8 | 2 | 5e-05 | 8 | 2 |

Table 52: Hyperparameters, namely the learning rate (LR), the total batch size (BS) and the number of epochs (NE) for the full and word-only models trained on NYT.

**Temporal Datasets** Hyperparameters are optimized for each combination of dataset and model using the oldest temporal training and development set. The same hyperparameters are then used for the remaining temporal splits for the dataset and model combination.

| Dataset | Model | LR | BS | EP |
|---|---|---|---|---|
| NER-TTC | bert-base-cased | 5e-05 | 2 | 10 |
| | bert-large-cased | 5e-06 | 2 | 6 |
| | distilbert-base-cased | 5e-05 | 2 | 10 |
| | roberta-base | 5e-05 | 2 | 10 |
| | roberta-large | 5e-06 | 2 | 6 |
| | distilroberta-base | 5e-05 | 2 | 10 |
| Truecasing-NYT | bert-base-cased | 5e-05 | 16 | 3 |
| | bert-large-cased | 5e-06 | 16 | 3 |
| | distilbert-base-cased | 5e-05 | 16 | 2 |
| | roberta-base | 5e-05 | 16 | 3 |
| | roberta-large | 5e-06 | 16 | 3 |
| | distilroberta-base | 5e-05 | 16 | 3 |
| Sentiment-Amazon | bert-base-cased | 5e-06 | 32 | 3 |
| | distilbert-base-cased | 5e-05 | 32 | 3 |
| | roberta-base | 5e-06 | 32 | 3 |
| | distilroberta-base | 5e-05 | 32 | 3 |
| Domain-NYT | bert-base-cased | 5e-05 | 32 | 2 |
| | distilbert-base-cased | 5e-05 | 32 | 2 |
| | roberta-base | 5e-05 | 32 | 4 |
| | distilroberta-base | 5e-05 | 32 | 3 |

Table 53: Hyperparameters, namely the learning rate (LR), the total batch size (BS) and the number of epochs (NE) for the models and datasets used to study temporal effects.

**Downsampled Datasets** For models trained on 100 sentences, the same learning rate and batch size are used as the model trained on the full training data. The number of epochs is optimized with values in {1, 2, 3, 4, 5, 8, 10, 20, 30, 40, 50, 60, 70}.

| | LR | BS | EP |
|---|---|---|---|
| CoNLL | 3e-05 | 16 | 30 |
| TTC-2014 | 5e-06 | 2 | 40 |
| arts | 3e-05 | 16 | 40 |
| business | 5e-05 | 8 | 60 |
| classified | 5e-05 | 4 | 30 |
| editorial | 5e-05 | 8 | 60 |
| foreign | 5e-05 | 4 | 30 |
| metropolitan | 5e-05 | 16 | 30 |
| national | 5e-05 | 16 | 40 |
| sports | 3e-05 | 8 | 40 |
| others | 5e-05 | 8 | 20 |
| nw | 3e-05 | 16 | 50 |
| bn | 5e-05 | 16 | 50 |
| bc | 3e-05 | 8 | 30 |
| tc | - | - | - |
| mz | 5e-05 | 8 | 50 |
| wb | 3e-05 | 8 | 60 |

Table 54: Hyperparameters, namely the learning rate (LR), the total batch size (BS) and the number of epochs (NE) for models trained on 100 sentences.

## APPENDIX B: Lists and Patterns

List of Honorifics and Regular Expressions used to caculate pattern occurence in Table 28.

**Honorifics** Following is the list of honorifics (mostly English, owing to the nature of the datasets) – Dr, Mr, Ms, Mrs, Mstr, Miss, Dr., Mr., Ms., Mrs., Mx., Mstr., Mister, Professor, Doctor, President, Senator, Judge, Governor, Officer, General, Nurse, Captain, Coach, Reverend, Rabbi, Ma'am, Sir, Father, Maestro, Madam, Colonel, Gentleman, Sire, Mistress, Lord, Lady, Esq, Excellency, Chancellor, Warden, Principal, Provost, Headmaster, Headmistress, Director, Regent, Dean, Chairman, Chairwoman, Chairperson, Pastor.

**Sports Scores** We use the following three regular expressions to determine if a sentence contains sports scores -

([0-9]+. )?([A-Za-z]+ ){1,3}([0-9]+ ){0,6}(([0-9]+)(?!))( 12)?( -)?

and

([A-Za-z]+ ){1,3}([0-9]+ ){1,3}([A-Za-z]+ ){1,3}([0-9]+ ){0,2}[0-9]

and

([A-Z]+ ){1,3}AT ([A-Z]+ ){1,2}[A-Z]+

## APPENDIX C: Links to Resources

Entity-switched Datasets (Chapter 3): https://github.com/oagarwal/entity-switched-ner

NYT-NER Dataset (Chapter 4): https://github.com/oagarwal/nyt-ner

Temporal Effects Data and Model Resources (Chapter 5): https://github.com/oagarwal/temporal-effects

Context-only models, MLM-kNN models and Data Augmentation Code (Chapter 7): https://github.com/oagarwal/ner-label-expansion

# BIBLIOGRAPHY


Abhishek Abhishek, Sanya Bathla Taneja, Garima Malik, Ashish Anand, and Amit Awekar. 2019. Fine-grained entity recognition with reduced false negatives and large type coverage. In *Automated Knowledge Base Construction (AKBC)*.

Oshin Agarwal, Heming Ge, Siamak Shakeri, and Rami Al-Rfou. 2021a. Knowledge graph based synthetic corpus generation for knowledge-enhanced language model pre-training. In *Proceedings of NAACL-HLT*, pages 3554–3565, Online. Association for Computational Linguistics.

Oshin Agarwal and Ani Nenkova. 2021. The utility and interplay of gazetteers and entity segmentation for named entity recognition in English. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 3990–4002, Online. Association for Computational Linguistics.

Oshin Agarwal and Ani Nenkova. 2022. Temporal effects on pre-trained models for language processing tasks. *Transactions of the Association for Computational Linguistics*, 10:904–921.

Oshin Agarwal and Ani Nenkova. 2023. Named entity recognition in a very homogeneous domain. In *Findings of the Association for Computational Linguistics: EACL 2023*, Online. Association for Computational Linguistics.

Oshin Agarwal, Sanjay Subramanian, Ani Nenkova, and Dan Roth. 2019. Evaluation of named entity coreference. In *Proceedings of the Second Workshop on Computational Models of Reference, Anaphora and Coreference*, pages 1–7, Minneapolis, USA. Association for Computational Linguistics.

Oshin Agarwal, Yinfei Yang, Byron C. Wallace, and Ani Nenkova. 2020. Entity-switched datasets: An approach to auditing the in-domain robustness of named entity recognition models. *arXiv preprint arXiv:2004.04123*.

Oshin Agarwal, Yinfei Yang, Byron C. Wallace, and Ani Nenkova. 2021b. Interpretability analysis for named entity recognition to understand system predictions and how they can improve. *Computational Linguistics*, 47(1):117–140.

Eugene Agichtein and Luis Gravano. 2000. Snowball: Extracting relations from large plain-text collections. In *Proceedings of the fifth ACM conference on Digital libraries*, pages 85–94. ACM.

Alan Akbik, Larysa Visengeriyeva, Johannes Kirschnick, and Alexander Löser. 2013. Effective selectional restrictions for unsupervised relation extraction. In *Proceedings of the Sixth International Joint Conference on Natural Language Processing*, pages 1312–1320, Nagoya, Japan.

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. Polyglot-ner: Massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining*, pages 586–594. SIAM.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017a. SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics.

Isabelle Augenstein, Leon Derczynski, and Kalina Bontcheva. 2017b. Generalisation in named entity recognition: A quantitative analysis. *Computer Speech & Language*, 44:61–83.

Dominic Balasuriya, Nicky Ringland, Joel Nothman, Tara Murphy, and James R. Curran. 2009. Named entity recognition in Wikipedia. In *Proceedings of the 2009 Workshop on The People's Web Meets NLP: Collaboratively Constructed Semantic Resources (People's Web)*, pages 10–18, Suntec, Singapore. Association for Computational Linguistics.

Michele Banko, Michael J Cafarella, Stephen Soderland, Matthew Broadhead, and Oren Etzioni. 2007. Open information extraction from the web. In *Ijcai*, volume 7, pages 2670–2676.

Iz Beltagy, Kyle Lo, and Arman Cohan. 2019. SciBERT: A pretrained language model for scientific text. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3615–3620, Hong Kong, China. Association for Computational Linguistics.

Oliver Bender, Franz Josef Och, and Hermann Ney. 2003. Maximum entropy models for named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 148–151.

Ruha Benjamin. 2019. *Race After Technology: Abolitionist Tools for the New Jim Code*. Polity.

Daniel M Bikel, Richard Schwartz, and Ralph M Weischedel. 1999. An algorithm that learns what's in a name. *Machine learning*, 34(1-3):211–231.

Johannes Bjerva, Wouter Kouw, and Isabelle Augenstein. 2020. Back to the future – temporal adaptation of text representations. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 7440–7447.

Su Lin Blodgett, Lisa Green, and Brendan T. O'Connor. 2016. Demographic dialectal variation in social media: A case study of african-american english. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing, EMNLP 2016, Austin, Texas, USA, November 1-4, 2016*, pages 1119–1130.

Kurt Bollacker, Colin Evans, Praveen Paritosh, Tim Sturge, and Jamie Taylor. 2008. Freebase: a collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, pages 1247–1250.

Stephanie Brandl and David Lassner. 2019. Times are changing: Investigating the pace of language change in diachronic word embeddings. In *Proceedings of the 1st International Workshop on Computational Approaches to Historical Language Change*, pages 146–150, Florence, Italy. Association for Computational Linguistics.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Joy Buolamwini and Timnit Gebru. 2018. Gender shades: Intersectional accuracy disparities in commercial gender classification. In *Conference on Fairness, Accountability and Transparency, FAT 2018, 23-24 February 2018, New York, NY, USA*, pages 77–91.

Shuguang Chen, Leonardo Neves, and Thamar Solorio. 2021. Mitigating temporal-drift: A simple approach to keep NER models crisp. In *Proceedings of the Ninth International Workshop on Natural Language Processing for Social Media*, pages 163–169, Online. Association for Computational Linguistics.

Emmanuele Chersoni, Adrià Torrens Urrutia, Philippe Blache, and Alessandro Lenci. 2018. Modeling violations of selectional restrictions with distributional semantics. In *Proceedings of the Workshop on Linguistic Complexity and Natural Language Processing*, pages 20–29.

Nancy A. Chinchor. 1998. Overview of MUC-7. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, 4:357–370.

Eunsol Choi, Omer Levy, Yejin Choi, and Luke Zettlemoyer. 2018. Ultra-fine entity typing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 87–96, Melbourne, Australia. Association for Computational Linguistics.

Michael Collins and Yoram Singer. 1999. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th international conference on Machine learning*, pages 160–167.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of machine learning research*, 12(Aug):2493–2537.

Silviu Cucerzan and David Yarowsky. 2002. Language independent ner using a unified model of internal and contextual evidence. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*.

Leyang Cui, Yu Wu, Jian Liu, Sen Yang, and Yue Zhang. 2021. Template-based named entity recognition using BART. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 1835–1845, Online. Association for Computational Linguistics.

Xiang Dai and Heike Adel. 2020. An analysis of simple data augmentation for named entity recognition. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3861–3867, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Xiang Dai, Sarvnaz Karimi, Ben Hachey, and Cecile Paris. 2019. Using similarity measures to select pretraining data for NER. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1460–1470, Minneapolis, Minnesota. Association for Computational Linguistics.

Cristian Danescu-Niculescu-Mizil, Robert West, Dan Jurafsky, Jure Leskovec, and Christopher Potts. 2013. No country for old members: user lifecycle and linguistic change in online communities. In *22nd International World Wide Web Conference, WWW '13, Rio de Janeiro, Brazil, May 13-17, 2013*, pages 307–318.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.

Leon Derczynski, Kalina Bontcheva, and Ian Roberts. 2016. Broad Twitter corpus: A diverse named entity recognition resource. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1169–1179, Osaka, Japan. The COLING 2016 Organizing Committee.

Leon Derczynski, Eric Nichols, Marieke van Erp, and Nut Limsopatham. 2017. Results of the WNUT2017 shared task on novel and emerging entity recognition. In *Proceedings of*

*the 3rd Workshop on Noisy User-generated Text*, Copenhagen, Denmark. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Bosheng Ding, Linlin Liu, Lidong Bing, Canasai Kruengkrai, Thien Hai Nguyen, Shafiq Joty, Luo Si, and Chunyan Miao. 2020. DAGA: Data augmentation with a generation approach for low-resource tagging tasks. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6045–6057, Online. Association for Computational Linguistics.

Rezarta Islamaj Doğan, Robert Leaman, and Zhiyong Lu. 2014. Ncbi disease corpus: a resource for disease name recognition and concept normalization. *Journal of biomedical informatics*, 47:1–10.

Rotem Dror, Gili Baumer, Segev Shlomov, and Roi Reichart. 2018. The hitchhiker's guide to testing statistical significance in natural language processing. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1383–1392, Melbourne, Australia. Association for Computational Linguistics.

P. Dury and P. Drouin. 2011. When terms disappear from a specialized lexicon: A semi-automatic investigation into necrology. *ICAME Journal*, pages 19–33.

Jacob Eisenstein. 2013. What to do about bad language on the internet. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 359–369, Atlanta, Georgia. Association for Computational Linguistics.

Jacob Eisenstein. 2019. Measuring and modeling language change. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Tutorials*, pages 9–14, Minneapolis, Minnesota. Association for Computational Linguistics.

Hady Elsahar and Matthias Gallé. 2019. To annotate or not? predicting performance drop under domain shift. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2163–2173, Hong Kong, China. Association for Computational Linguistics.

Ali Emami, Paul Trichelair, Adam Trischler, Kaheer Suleman, Hannes Schulz, and Jackie Chi Kit Cheung. 2019. The KnowRef coreference corpus: Removing gender and number cues for difficult pronominal anaphora resolution. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3952–3961, Florence, Italy. Association for Computational Linguistics.

Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S Weld, and Alexander Yates. 2005. Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134.

Matan Eyal, Tal Baumel, and Michael Elhadad. 2019. Question answering as an automatic evaluation metric for news article summarization. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3938–3948, Minneapolis, Minnesota. Association for Computational Linguistics.

Tom Fawcett. 2003. " in vivo" spam filtering: a challenge problem for kdd. *ACM SIGKDD Explorations Newsletter*, 5(2):140–148.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 363–370, Ann Arbor, Michigan. Association for Computational Linguistics.

Francesc Ribas Framis. 1994. An experiment on learning appropriate selectional restrictions from a parsed corpus. In *COLING 1994 Volume 2: The 15th International Conference on Computational Linguistics*, pages 769–774.

Hege Fromreide, Dirk Hovy, and Anders Søgaard. 2014. Crowdsourcing and annotating NER for Twitter #drift. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2544–2547, Reykjavik, Iceland. European Language Resources Association (ELRA).

Jinlan Fu, Pengfei Liu, and Graham Neubig. 2020a. Interpretable multi-dataset evaluation for named entity recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6058–6069, Online. Association for Computational Linguistics.

Jinlan Fu, Pengfei Liu, Qi Zhang, and Xuanjing Huang. 2020b. Rethinking generalization of neural models: A named entity recognition case study.

William A. Gale, Kenneth Ward Church, and David Yarowsky. 1995. Discrimination decisions for 100,000-dimensional spaces. *Annals of Operations Research*, 55:429–450.

Octavian-Eugen Ganea and Thomas Hofmann. 2017. Deep joint entity disambiguation with local neural attention. In *Proceedings of the 2017 Conference on Empirical Methods in*

*Natural Language Processing*, pages 2619–2629, Copenhagen, Denmark. Association for Computational Linguistics.

Abbas Ghaddar and Philippe Langlais. 2018a. Transforming Wikipedia into a large-scale fine-grained entity type corpus. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Abbas Ghaddar, Philippe Langlais, Ahmad Rashid, and Mehdi Rezagholizadeh. 2021. Context-aware adversarial training for name regularity bias in named entity recognition. *Transactions of the Association for Computational Linguistics*, 9:586–604.

Abbas Ghaddar and Phillippe Langlais. 2017. WiNER: A Wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Abbas Ghaddar and Phillippe Langlais. 2018b. Robust lexical features for improved neural network named-entity recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1896–1907, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Dan Gillick, Nevena Lazic, Kuzman Ganchev, Jesse Kirchner, and David Huynh. 2014. Context-dependent fine-grained entity type tagging. *arXiv preprint arXiv:1412.1820*.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

William L. Hamilton, Jure Leskovec, and Dan Jurafsky. 2016. Diachronic word embeddings reveal statistical laws of semantic change. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1489–1501, Berlin, Germany. Association for Computational Linguistics.

Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. 2004. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 415–422, Barcelona, Spain.

Xuanli He, Islam Nassar, Jamie Kiros, Gholamreza Haffari, and Mohammad Norouzi. 2022. Generate, annotate, and learn: NLP with synthetic text. *Transactions of the Association for Computational Linguistics*, 10:826–842.

Yu He, Jianxin Li, Yangqiu Song, Mutian He, and Hao Peng. 2018. Time-evolving text classification with deep neural networks. In *Proceedings of the 27th International Joint Conference on Artificial Intelligence*, IJCAI'18, page 2241–2247. AAAI Press.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Valentin Hofmann, Janet Pierrehumbert, and Hinrich Schütze. 2021. Dynamic contextualized word embeddings. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 6970–6984, Online. Association for Computational Linguistics.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: The 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 57–60, New York City, USA. Association for Computational Linguistics.

Xiaolei Huang and Michael J. Paul. 2018. Examining temporality in document classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 694–699, Melbourne, Australia. Association for Computational Linguistics.

Xiaolei Huang and Michael J. Paul. 2019. Neural temporality adaptation for document classification: Diachronic word embeddings and domain adaptation models. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4113–4123, Florence, Italy. Association for Computational Linguistics.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Divyansh Kaushik, Eduard Hovy, and Zachary C. Lipton. 2019. Learning the difference that makes a difference with counterfactually-augmented data.

Jun'ichi Kazama and Kentaro Torisawa. 2007a. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707, Prague, Czech Republic. Association for Computational Linguistics.

Jun'ichi Kazama and Kentaro Torisawa. 2007b. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 698–707, Prague, Czech Republic. Association for Computational Linguistics.

Hyunjae Kim, Jaehyo Yoo, Seunghyun Yoon, Jinhyuk Lee, and Jaewoo Kang. 2022. Simple

questions generate named entity recognition datasets. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 6220–6236, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

J-D Kim, Tomoko Ohta, Yuka Tateisi, and Jun'ichi Tsujii. 2003. Genia corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl_1):i180–i182.

Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, et al. 2015. The chemdner corpus of chemicals and drugs and its annotation principles. *Journal of cheminformatics*, 7(1):1–17.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, page 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 260–270, San Diego, California. Association for Computational Linguistics.

A. Lazaridou, A. Kuncoro, E. Gribovskaya, Devang Agrawal, Adam Liska, Tayfun Terzi, Mai Gimenez, Cyprien de Masson d'Autume, Sebastian Ruder, Dani Yogatama, Kris Cao, Tomás Kociský, Susannah Young, and P. Blunsom. 2021. Mind the gap: Assessing temporal generalization in neural language models. *Advances in Neural Information Processing Systems*, 34.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Xiaoya Li, Jingrong Feng, Yuxian Meng, Qinghong Han, Fei Wu, and Jiwei Li. 2020. A unified MRC framework for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 5849–5859, Online. Association for Computational Linguistics.

Chen Liang, Yue Yu, Haoming Jiang, Siawpeng Er, Ruijia Wang, Tuo Zhao, and Chao Zhang. 2020. Bond: Bert-assisted open-domain named entity recognition with distant supervision. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '20, page 1054–1064, New York, NY, USA. Association for Computing Machinery.

Wenhui Liao and Sriharsha Veeramachaneni. 2009. A simple semi-supervised algorithm for

named entity recognition. In *Proceedings of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing*, SemiSupLearn '09, pages 58–65, Stroudsburg, PA, USA. Association for Computational Linguistics.

Bill Yuchen Lin, Dong-Ho Lee, Ming Shen, Ryan Moreno, Xiao Huang, Prashant Shiralkar, and Xiang Ren. 2020. TriggerNER: Learning with entity triggers as explanations for named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8503–8511, Online. Association for Computational Linguistics.

Xiao Ling and Daniel Weld. 2012. Fine-grained entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 26(1):94–100.

Lucian Vlad Lita, Abe Ittycheriah, Salim Roukos, and Nanda Kambhatla. 2003. tRuEcasIng. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 152–159, Sapporo, Japan. Association for Computational Linguistics.

Andy T Liu, Wei Xiao, Henghui Zhu, Dejiao Zhang, Shang-Wen Li, and Andrew Arnold. 2022. Qaner: Prompting question answering models for few-shot named entity recognition. *arXiv preprint arXiv:2203.01543*.

Tianyu Liu, Jin-Ge Yao, and Chin-Yew Lin. 2019a. Towards improving neural named entity recognition with gazetteers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5301–5307, Florence, Italy. Association for Computational Linguistics.

Y. Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, M. Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019b. Roberta: A robustly optimized bert pretraining approach. *ArXiv*, abs/1907.11692.

Yijin Liu, Fandong Meng, Jinchao Zhang, Jinan Xu, Yufeng Chen, and Jie Zhou. 2019c. GCDT: A global context enhanced deep transition architecture for sequence labeling. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2431–2441, Florence, Italy. Association for Computational Linguistics.

Teng Long, Ryan Lowe, Jackie Chi Kit Cheung, and Doina Precup. 2016. Leveraging lexical resources for learning entity embeddings in multi-relational data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 112–117, Berlin, Germany. Association for Computational Linguistics.

Jan Lukes and Anders Søgaard. 2018. Sentiment analysis under temporal shift. In *Proceedings of the 9th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, pages 65–71, Brussels, Belgium. Association for Computational Linguistics.

Jie Ma, Miguel Ballesteros, Srikanth Doss, Rishita Anubhai, Sunil Mallya, Yaser Al-Onaizan, and Dan Roth. 2022. Label semantics for few shot named entity recognition. In *Findings*

*of the Association for Computational Linguistics: ACL 2022*, pages 1956–1971, Dublin, Ireland. Association for Computational Linguistics.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1064–1074, Berlin, Germany. Association for Computational Linguistics.

Yukun Ma, Erik Cambria, and Sa Gao. 2016. Label embedding for zero-shot fine-grained named entity typing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 171–180, Osaka, Japan. The COLING 2016 Organizing Committee.

Simone Magnolini, Valerio Piccioni, Vevake Balaraman, Marco Guerini, and Bernardo Magnini. 2019. How to use gazetteers for entity recognition with neural models. In *Proceedings of the 5th Workshop on Semantic Deep Learning (SemDeep-5)*, pages 40–49, Macau, China. Association for Computational Linguistics.

Gretchen McCulloch. 2020. *Because internet: Understanding the new rules of language.* Riverhead Books.

David McDonald. 1993. Internal and external evidence in the identification and semantic categorization of proper names. In *Acquisition of Lexical Knowledge from Text.*

Xue Mengge, Bowen Yu, Zhenyu Zhang, Tingwen Liu, Yue Zhang, and Bin Wang. 2020. Coarse-to-Fine Pre-training for Named Entity Recognition. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6345–6354, Online. Association for Computational Linguistics.

Andrei Mikheev, Marc Moens, and Claire Grover. 1999. Named entity recognition without gazetteers. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–8, Bergen, Norway. Association for Computational Linguistics.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Proceedings of the 26th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'13, page 3111–3119, Red Hook, NY, USA. Curran Associates Inc.

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

Scott Miller, Jethran Guinness, and Alex Zamanian. 2004. Name tagging with word clusters and discriminative training. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics: HLT-NAACL 2004*, pages 337–342.

Einat Minkov, Richard C. Wang, and William W. Cohen. 2005. Extracting personal names

from email: Applying named entity recognition to informal text. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, pages 443–450, Vancouver, British Columbia, Canada. Association for Computational Linguistics.

Shikhar Murty, Patrick Verga, Luke Vilnis, Irena Radovanovic, and Andrew McCallum. 2018. Hierarchical losses and new resources for fine-grained entity typing and linking. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 97–109, Melbourne, Australia. Association for Computational Linguistics.

David Nadeau, Peter D Turney, and Stan Matwin. 2006. Unsupervised named-entity recognition: Generating gazetteers and resolving ambiguity. In *Conference of the Canadian society for computational studies of intelligence*, pages 266–277. Springer.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Jianmo Ni, Jiacheng Li, and Julian McAuley. 2019. Justifying recommendations using distantly-labeled reviews and fine-grained aspects. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 188–197, Hong Kong, China. Association for Computational Linguistics.

Chikashi Nobata, Satoshi Sekine, Hitoshi Isahara, and Ralph Grishman. 2002. Summarization system integrated with named entity tagging and IE pattern discovery. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, pages 1742–1745, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Joel Nothman, James R. Curran, and Tara Murphy. 2008. Transforming Wikipedia into named entity training data. In *Proceedings of the Australasian Language Technology Association Workshop 2008*, pages 124–132, Hobart, Australia.

Naoaki Okazaki. 2007. Crfsuite: a fast implementation of conditional random fields (crfs).

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014a. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014b. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Matthew Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 2227–2237, New Orleans, Louisiana. Association for Computational Linguistics.

Vinodkumar Prabhakaran, Ben Hutchinson, and Margaret Mitchell. 2019. Perturbation sensitivity analysis to detect unintended model biases. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5740–5745, Hong Kong, China. Association for Computational Linguistics.

Sameer S. Pradhan and Nianwen Xue. 2009. OntoNotes: The 90% solution. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Tutorial Abstracts*, pages 11–12, Boulder, Colorado. Association for Computational Linguistics.

Duangmanee Putthividhya and Junling Hu. 2011. Bootstrapped named entity recognition for product attribute extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1557–1567, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2022. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(1).

Jonathan Raiman and John Miller. 2017. Globally normalized reader. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1059–1069, Copenhagen, Denmark. Association for Computational Linguistics.

Lev Ratinov and Dan Roth. 2009. Design challenges and misconceptions in named entity recognition. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning (CoNLL-2009)*, pages 147–155, Boulder, Colorado. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of LSTM-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348, Copenhagen, Denmark. Association for Computational Linguistics.

Philip Stuart Resnik. 1993. Selection and information: A class-based approach to lexical relationships. *IRCS Technical Reports Series*, page 200.

Shruti Rijhwani and Daniel Preotiuc-Pietro. 2020. Temporally-informed analysis of named entity recognition. In *Proceedings of ACL*, pages 7605–7617, Online. Association for Computational Linguistics.

Ellen Riloff and Rosie Jones. 1999. Learning dictionaries for information extraction by multi-level bootstrapping. In *Proceedings of the Sixteenth National Conference on Artificial Intelligence and the Eleventh Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, AAAI '99/IAAI '99, page 474–479, USA. American Association for Artificial Intelligence.

Alan Ritter, Sam Clark, Mausam, and Oren Etzioni. 2011. Named entity recognition in tweets: An experimental study. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Alex Rosenfeld and Katrin Erk. 2018. Deep neural models of semantic shift. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 474–484, New Orleans, Louisiana. Association for Computational Linguistics.

Guy D. Rosin and Kira Radinsky. 2022. Temporal attention for language models. In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1498–1508, Seattle, United States. Association for Computational Linguistics.

Paul Röttger and Janet Pierrehumbert. 2021. Temporal adaptation of BERT and performance on downstream document classification: Insights from social media. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2400–2412, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Evan Sandhaus. 2008. The new york times annotated corpus. *Linguistic Data Consortium, Philadelphia*, 6(12):e26752.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *arXiv preprint arXiv:1910.01108*.

Satoshi Sekine, Kiyoshi Sudo, and Chikashi Nobata. 2002. Extended named entity hierarchy. In *Proceedings of the Third International Conference on Language Resources and Evaluation (LREC'02)*, Las Palmas, Canary Islands - Spain. European Language Resources Association (ELRA).

Dominic Seyler, Tatiana Dembelova, Luciano Del Corro, Johannes Hoffart, and Gerhard Weikum. 2018. A study of the importance of external knowledge in the named entity recognition task. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 241–246, Melbourne, Australia. Association for Computational Linguistics.

Anders Søgaard, Sebastian Ebert, Jasmijn Bastings, and Katja Filippova. 2021. We need to talk about random splits. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 1823–1832.

Chan Hee Song, Dawn Lawrie, Tim Finin, and James Mayfield. 2020. Improving neural named entity recognition with gazetteers. In *Proceedings of the 33rd International FLAIRS Conference*.

Benjamin Strauss, Bethany Toma, Alan Ritter, Marie-Catherine de Marneffe, and Wei Xu. 2016. Results of the WNUT16 named entity recognition shared task. In *Proceedings of the 2nd Workshop on Noisy User-generated Text (WNUT)*, pages 138–144, Osaka, Japan. The COLING 2016 Organizing Committee.

Sanjay Subramanian and Dan Roth. 2019. Improving generalization in coreference resolution via adversarial training. In *Proceedings of the Eighth Joint Conference on Lexical and Computational Semantics (*SEM 2019)*, pages 192–197, Minneapolis, Minnesota. Association for Computational Linguistics.

Fabian M. Suchanek, Gjergji Kasneci, and Gerhard Weikum. 2007. Yago: A core of semantic knowledge. In *Proceedings of the 16th International Conference on World Wide Web*, WWW '07, page 697–706, New York, NY, USA. Association for Computing Machinery.

Partha Pratim Talukdar, Thorsten Brants, Mark Liberman, and Fernando Pereira. 2006. A context pattern induction method for named entity extraction. In *Proceedings of the Tenth Conference on Computational Natural Language Learning (CoNLL-X)*, pages 141–148, New York City. Association for Computational Linguistics.

Simone Tedeschi and Roberto Navigli. 2022. MultiNERD: A multilingual, multi-genre and fine-grained dataset for named entity recognition (and disambiguation). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 801–812, Seattle, United States. Association for Computational Linguistics.

Erik F. Tjong Kim Sang and Fien De Meulder. 2003. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, pages 142–147.

Jing Wang, Mayank Kulkarni, and Daniel Preotiuc-Pietro. 2020. Multi-domain named entity recognition with genre-aware and agnostic inference. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8476–8488, Online. Association for Computational Linguistics.

Uriel Weinreich, W. Labov, and Marvin Herzog. 1968. Empirical foundations for a theory of language change.

Derry Tanti Wijaya and Reyyan Yeniterzi. 2011. Understanding semantic change of words over centuries. In *Proceedings of the 2011 International Workshop on DETecting and Exploiting Cultural DiversiTy on the Social Web*, DETECT '11, page 35–40, New York, NY, USA. Association for Computing Machinery.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam

Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Y. Wu, Mike Schuster, Z. Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, M. Krikun, Yuan Cao, Q. Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, Taku Kudo, H. Kazawa, K. Stevens, G. Kurian, Nishant Patil, W. Wang, C. Young, J. Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, G. S. Corrado, Macduff Hughes, and J. Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Vikas Yadav and Steven Bethard. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 2145–2158, Santa Fe, New Mexico, USA. Association for Computational Linguistics.

Eunsuk Yang, Young-Bum Kim, Ruhi Sarikaya, and Yu-Seop Kim. 2016. Drop-out conditional random fields for Twitter with huge mined gazetteer. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 282–288, San Diego, California. Association for Computational Linguistics.

Yi Yang and Arzoo Katiyar. 2020. Simple and effective few-shot named entity recognition with structured nearest neighbor learning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6365–6375, Online. Association for Computational Linguistics.

Zhixiu Ye and Zhen-Hua Ling. 2018. Hybrid semi-Markov CRF for neural sequence labeling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 235–240, Melbourne, Australia. Association for Computational Linguistics.

Wenpeng Yin, Jamaal Hay, and Dan Roth. 2019. Benchmarking zero-shot text classification: Datasets, evaluation and entailment approach. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3914–3923, Hong Kong, China. Association for Computational Linguistics.

Dani Yogatama, Daniel Gillick, and Nevena Lazic. 2015. Embedding methods for fine grained entity type classification. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 291–296, Beijing, China. Association for Computational Linguistics.

Mohamed Amir Yosef, Sandro Bauer, Johannes Hoffart, Marc Spaniol, and Gerhard

Weikum. 2012. HYENA: Hierarchical type classification for entity names. In *Proceedings of COLING 2012: Posters*, pages 1361–1370, Mumbai, India. The COLING 2012 Organizing Committee.

Zheng Yuan and Doug Downey. 2018. Otyper: A neural architecture for open named entity typing. In *AAAI Conference on Artificial Intelligence*.

Yuan Zhang, Jason Riesa, Daniel Gillick, Anton Bakalov, Jason Baldridge, and David Weiss. 2018. A fast, compact, accurate model for language identification of codemixed text. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 328–337.

Liang Zhou, Miruna Ticrea, and Eduard Hovy. 2004. Multi-document biography summarization. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 434–441, Barcelona, Spain. Association for Computational Linguistics.

Ran Zhou, Xin Li, Ruidan He, Lidong Bing, Erik Cambria, Luo Si, and Chunyan Miao. 2022. MELM: Data augmentation with masked entity language modeling for low-resource NER. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2251–2262, Dublin, Ireland. Association for Computational Linguistics.