Named Entity Recognition in a Very Homogeneous Domain

Oshin Agarwal
University of Pennsylvania
oagarwal@seas.upenn.edu

Ani Nenkova
Adobe Research
nenkova@adobe.com

Abstract

Machine Learning models have lower accuracy when tested on out-of-domain data. Developing models that perform well on several domains or can be quickly adapted to a new domain is an important research area. Domain, however, is a vague term, that can refer to any aspect of data such as language, genre, source and structure. We consider a very homogeneous source of data, specifically sentences from news articles from the same newspaper in English, and collect a dataset of such "in-domain" sentences annotated with named entities. We find that even in such a homogeneous domain, the performance of named entity recognition models varies significantly across news topics. Selection of diverse data, as we demonstrate, is crucial even in a seemingly homogeneous domain.

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.

2 Dataset

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

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[1], 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.

[1] desk in NYT newsroom that produced the article

Findings of the Association for Computational Linguistics: EACL 2023, pages 1850–1855

2.2 Data Annotation

The selected sentences are labeled with person (*PER*), location (*LOC*) and organization (*ORG*) tags on Upwork[2], 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 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.

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 1.

	# 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 1: Dataset Statistics

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.

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 2. 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 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 1) for the test sub-domains of classifieds

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

	InD	OOD					
		mean	median	min		max	
	F1	F1	F1	F1	trn-d	F1	trn-d
a	92.1	86.9	88.3	78.4	c	89.7	m
b	95.7	88.2	90.9	72.0	c	93.2	m
c	94.7	77.7	76.7	67.1	f	90.4	e
e	96.4	88.7	93.0	67.0	c	94.6	n
f	96.9	87.5	92.5	64.2	c	93.9	n
m	95.0	89.2	90.8	78.4	c	92.8	n
n	96.2	90.9	93.8	79.8	c	94.9	m
s	94.8	81.0	81.0	77.9	n	84.6	a
o	92.0	87.4	89.0	76.7	c	90.8	m

Table 2: 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.

	InD	OOD					
		mean	median	min		max	
	F1	F1	F1	F1	tst-d	F1	tst-d
a	92.1	87.7	90.4	73.2	c	91.6	b
b	95.7	88.7	90.0	78.1	c	94.0	e
c	94.7	74.3	77.5	64.2	f	79.8	n
e	96.4	89.4	90.3	80.4	s	94.2	n
f	96.9	85.8	88.8	67.1	c	93.6	n
m	95.0	90.6	92.0	84.0	s	94.9	n
n	96.2	89.2	91.1	77.9	s	94.6	e
s	94.8	82.4	85.1	68.8	c	86.6	n
o	92.0	89.2	92.3	75.3	c	94.1	e

Table 3: 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.

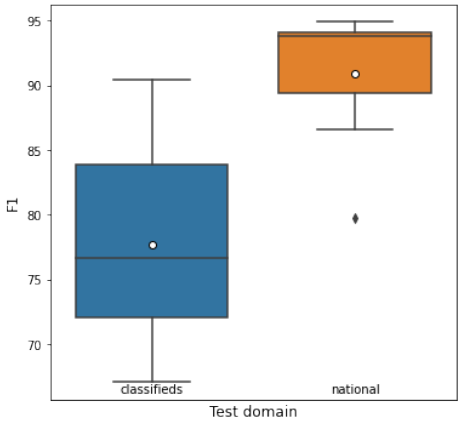

Figure 1: Box plot for two test sub-domains (classifieds and national) showing the range of F1 with training on OOD sub-domains

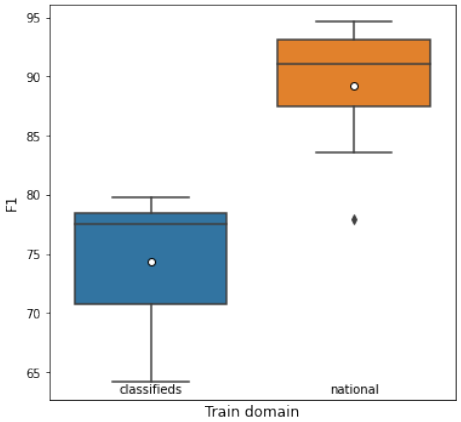

Figure 2: Box plot for two training sub-domains (classified and national), showing the range of F1 when tested on these as OOD sub-domains.

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.

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 3. 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 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 1) for the training sub-domains of classified and national. Depending

Domain	Sentence
Classifieds	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 4: Example of sentences by sub-domain

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 4. 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 the training data, labeling them correctly is unlikely.

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.[3] 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 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 5. 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 5 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[4] do not exist. Derczynski et al. (2016) selects tweets from different countries and different types of user accounts for linguistic variations and topics. They also account for temporal variation taking tweets from different years, months, weeks and days.

We build models with this random sentence selection scheme. We first downsample the data such that it follows the same distribution of sub-domains as the NYT corpus with 20 years of articles. This results in 10,500 training and 4,494 development sentences with 14% arts, 11% business, 3% classi-

[3] 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.

[4] 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 5: 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.

fied, 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. With models trained on this dataset, the average F1 is almost the same as in-subdomain training (col Rndm).

5 Conclusion

Perform fine-grained inspection of data even when it seems that the domain is homogeneous, and perform training data selection at the sentence level rather than the document level.

6 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[5] (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.

References

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.

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.

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.*

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.

Hal Daumé III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of*

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

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.

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.

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.

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.

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.

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

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.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, R'emi Louf, Morgan Funtowicz, and Jamie Brew. 2019. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771.

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.

A Hyperparameters and Infrastructure

Hyperparameters are optimized via grid search over the learning rate (3e-05, 5e-06, 5e-06), batch size (2, 4, 8, 16, 32) and number of epochs (1, 2, 3, 4, 5) on each sub-domain. Models were fine-tuned using the implementation in HuggingFace (Wolf et al., 2019) on 2 V100 GPUs. The training time for models varies by the sub-domain and hyperparameters, and is typically 10-20 min. The best checkpoint on the development set is selected.

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

Table 6: Hyperparameters, namely the learning rate, the total batch size and the number of epochs.

B Full Evaluation

Test	Training Domain								
	a	b	c	e	f	m	n	s	o
a	92.1	88.7	78.4	87.9	87.1	89.7	88.7	85.7	89.2
b	91.6	95.7	72.0	90.2	90.0	93.2	91.9	84.7	92.0
c	73.2	78.1	94.7	90.4	67.1	84.7	83.6	68.8	75.3
e	91.0	94.0	67.0	96.4	92.1	94.3	94.6	82.1	94.1
f	90.4	92.9	64.2	92.1	96.9	93.3	93.9	80.0	93.1
m	90.4	91.0	78.4	91.4	90.6	95.0	92.8	86.2	92.5
n	90.3	94.0	79.8	94.2	93.6	94.9	96.2	86.6	94.1
s	84.6	81.7	78.3	80.4	78.1	84.0	77.9	94.8	82.9
o	90.1	89.0	76.7	88.9	87.6	90.8	90.3	85.5	92.0

Table 7: F1 on model trained on each sub-domain on each of the sub-domains