Evaluating Pretrained Transformer-based Models on the Task of Fine-Grained Named Entity Recognition

Cedric Lothritz
University of Luxembourg
cedric.lothritz@uni.lu

Kevin Allix
University of Luxembourg
kevin.allix@uni.lu

Lisa Veiber
University of Luxembourg
lisa.veiber@uni.lu

Tegawendé F. Bissyandé
University of Luxembourg
tegawende.bissyande@uni.lu

Jacques Klein
University of Luxembourg
jacques.klein@uni.lu

Abstract

Named Entity Recognition (NER) is a fundamental Natural Language Processing (NLP) task and has remained an active research field. In recent years, transformer models and more specifically the BERT model developed at Google revolutionised the field of NLP. While the performance of transformer-based approaches such as BERT has been studied for NER, there has not yet been a study for the fine-grained Named Entity Recognition (FG-NER) task. In this paper, we compare three transformer-based models (BERT, RoBERTa, and XLNet) to two non-transformer-based models (CRF and BiLSTM-CNN-CRF). Furthermore, we apply each model to a multitude of distinct domains. We find that transformer-based models incrementally outperform the studied non-transformer-based models in most domains with respect to the F1 score. Furthermore, we find that the choice of domain significantly influenced the performance regardless of the respective data size or the model chosen.

1 Introduction

Named Entity Recognition (NER) is part of the fundamental tasks in Natural Language Processing (NLP). The main objective of NER is to detect and classify proper names (named entities) in a free text. Typically, named entities can be subdivided into four broad categories: **persons**, i.e., first and last names, **locations** such as countries or landscapes, **organisations** such as companies or political parties, and **miscellaneous entities** which serves as a catch-all category for other named entities such as brands, meals, or social events. NER is an active research field and state-of-the-art solutions such as spaCy[1], flair (Akbik et al., 2018), and Primer[2] manage to achieve near-human performance. However, classical NER (which we refer to as coarse-grained NER in this paper) models typically distinguish between only a small number of entity types, usually fewer than a dozen distinct categories.

While this kind of shallow classification is sufficient for many applications, there are industrial use-cases in which more precise information is necessary such as financial documents processing in the banking and finance context. For instance, application forms for a business loan are usually supplied with several supporting textual documents. These can contain the names of different types of persons, such as the owner or the CEO of the applying company, the contact person(s) at the issuing bank, finance analysts, or lawyers. The same is true for organisation names such as the name of the issuing bank, a government agency, or the name of the applying company or third-party companies. It is necessary to not only detect entity names, but to also *qualify* and differentiate between various entity types. Indeed, in many contexts the actual name of an entity is important only if it can be associated to a *role*, or any other relevant *quality*. In the banking and finance world for example, the strict regulatory requirements cannot be satisfied with just a list of *who* is involved; knowing *how* entities are involved is a necessity.

The term "Fine-Grained Named Entity Recognition" (FG-NER) was first coined by Fleischman and Hovy (2002). It describes a subtask of NER, where the objective remains the same as standard NER,

[1] https://spacy.io

[2] https://primer.ai/blog/a-new-state-of-the-art-for-named-entity-recognition/

Proceedings of the 28th International Conference on Computational Linguistics, pages 3750–3760
Barcelona, Spain (Online), December 8-13, 2020

but where the number of entity types is considerably higher. In extreme cases, FG-NER models such as the **fi**ne-**g**rained **e**ntity **r**ecognizer (FIGER) (Ling and Weld, 2012) are able to distinguish between more than 100 distinct labels.

Conditional Random Field (CRF) models (Lafferty et al., 2001) have been popular for numerous sequence-to-sequence tasks such as NER. They perform reasonably well and can serve as a baseline for the task of FG-NER.

In a previous study, Mai et al. (2018) compared the performance of several FG-NER approaches for the English and Japanese languages. They found that the BiLSTM-CNN-CRF model devised by Ma and Hovy (2016) combined with gazetteers performed the best in terms of F1 score for the English language. They also found that BiLSTM-CNN-CRF performed well without the use of gazetteers. In fact, among the models that did not make use of gazetteers, BiLSTM-CNN-CRF achieved the highest F1 score. In 2017, the introduction of the transformer model (Vaswani et al., 2017) revolutionised the NLP landscape and led to a number of novel language modeling approaches which manage to outperform state-of-the-art models in numerous tasks. In 2018, Devlin et al. (2019) developed the **B**idirectional **E**ncoder **R**epresentations from **T**ransformers (BERT) model, a powerful language modeling technique which is considered as one of the most significant breakthroughs in NLP in recent memory. BERT models are pretrained on Masked Language Modeling (MLM) and Next Sentence Prediction (NSP) tasks. Devlin et al. (2019) fine-tuned the resulting models on several fundamental NLP tasks such as the GLUE language understanding tasks (Wang et al., 2018), the SQuAD question answering task (Rajpurkar et al., 2016), and the SWAG Common Sense Inference task (Zellers et al., 2018), for which BERT manages to achieve state-of-the-art performances. Furthermore, Devlin et al. (2019) reported an F1 score of 92.8% when fine-tuned on the CoNLL-2003 dataset for NER (Sang and De Meulder, 2003), achieving similar results as state-of-the-art models such as Contextual String Embeddings (Akbik et al., 2018) and ELMo Embeddings (Peters et al., 2017).

Improving on the BERT model, Liu et al. (2019) at Facebook AI[3] developed a **R**obustly **o**ptimized **BERT a**pproach (RoBERTa). They claim that the standard BERT models were undertrained and proposed a new version of BERT that was trained for a longer time, on longer sequences, on more data, and with larger batches. Furthermore, they trained only on the MLM task and with dynamic changes of the masking patterns applied to training data. BERT's pretraining steps was performed on the same dataset using the same masked locations for the entire MLM task. RoBERTa mitigated that problem by duplicating their dataset ten times, and using different masking patterns for each duplicate. They report that fine-tuned models derived from RoBERTa either matched or improved on BERT models in terms of performance, although they did not perform tests specifically on the NER task.

2019 also saw an attempt to solve the shortcomings of BERT in terms of the training approach. Yang et al. (2019) presented XLNet. During the MLM pretraining task of BERT, a special [MASK] token is introduced in the training set. According to (Yang et al., 2019), BERT models neglect dependencies between the masked tokens. Furthermore, this token is absent in the fine-tuning tasks, resulting in a pretrain/fine-tune discrepancy. XLNet avoids this shortcoming as it does not mask its tokens, and instead permutes the order of token predictions. Yang et al. (2019) reports that XLNet outperforms BERT in 20 NLP tasks, specifically language understanding, reading comprehension, text classification and document ranking tasks. They do not report any results on sequence-to-sequence tasks like NER.

While BERT, RoBERTa, and XLNet (which we refer to as transformer-based models throughout the paper) achieve state-of-the-art performances in numerous Natural Language Understanding (NLU) tasks, we observe a lack of research in the area of FG-NER. In this paper, we present an empirical study of the performance of FG-NER approaches derived from a pretrained BERT, a pretrained RoBERTa, and a pretrained XLNet model as well as a comparison to a simple CRF model and the model presented by Ma and Hovy (2016). Furthermore, we apply these approaches to a large number of distinct domains, with varying numbers of data samples and entity categories.

Specifically, we will address the following research questions:

- RQ1: Do transformer-based models outperform the state-of-the-art model for the FG-NER task?

[3] https://ai.facebook.com/

ID	domain	#sentences	#words	#named entities before removal	#entity types after removal	#entity types
1	physics	68	1916	144	6	5
2	fashion	1043	27 598	2182	68	20
3	finance	1723	42 834	4121	75	24
4	exhibitions	1829	40 162	2960	131	34
5	meteorology	2838	69 551	7659	92	32
...	...	...	...	...	...	..
30	food	41 160	1 034 233	100 445	415	50
31	media	49 714	1 269 641	142 084	959	50
32	biology	53 042	1 248 434	142 084	246	50
33	travel	59 965	1 467 691	152 712	750	50
34	business	68 244	1 688 935	182 306	1009	50
...	...	...	...	...	...	...
45	government	331 720	8 380 706	1 170 947	1182	51
46	film	430 693	9 557 747	1 720 973	1134	51
47	music	441 220	10 116 628	1 684 479	918	50
48	people	442 683	11 452 451	1 762 255	1825	50
49	location	443 646	12 525 545	1 472 198	1603	50

Table 1: Statistics for a selection of datasets

- RQ2: What are the strengths, weaknesses, and trade-offs of each investigated model?
- RQ3: How does the choice of the domain influence the performance of the models?

We use the EWNERTC dataset published by Sahin et al. (2017a), containing roughly 7 million data samples in 49 different domains. To the best of our knowledge, our study is the first aiming to precisely evaluate the performance of these existing approaches on the FG-NER task.

2 Experimental Setup

In this section, we present the dataset used in this study and we introduce the different models that we compare against each other.

2.1 Dataset

For this study, we apply the selected models to the English Wikipedia Named Entity Recognition and Text Categorization (EWNERTC) dataset [4] published by Sahin et al. (2017b). It is a collection of automatically categorised and annotated sentences from Wikipedia articles. The original dataset consists of roughly 7 million annotated sentences, divided into 49 separate domains. These 49 domains vary significantly in overall size and number of entity types. The *physics* domain is the smallest subset with 68 sentences, 144 entities and merely 6 distinct entity types. In contrast, the *location* domain is the largest subset with 443 646 sentences, 1 472 198 entities, and 1603 types. Table 1 contains statistics for a small selection of domains.[5] *Physics*, *fashion*, *finance*, *exhibitions*, and *meteorology* are the five smallest sets, consisting of fewer than 3000 sentences each. *Food*, *media*, *biology*, *travel*, and *business* are medium-sized sets, comprising between 40 000 and 70 000 sentences. Finally, *government*, *film*, *music*, *people*, and *location* are the largest sets with more than 300 000 sentences each.

It is noteworthy that the *physics* dataset is an obvious outlier in terms of size (since the second smallest dataset is the *fashion* dataset, which contains an order of magnitude more sentences). It is possible that the size of the *physics* subset is too small to produce meaningful results.

For this study, the number of entity types was drastically reduced. This measure was taken for two reasons: most entity types appear only a few times in any given subset. Furthermore, the training time for CRF models tends to explode when dealing with a high number of entity types according to Mai et al. (2018). We limited the number of entity types per domain to the top 50 and, if necessary, added a *miscellaneous* type as a catch-all for all remaining named entities.

[4] https://data.mendeley.com/datasets/cdcztymf4k/1

[5] Link to the full table: https://github.com/lothritz/FG-NER-data-statistics/blob/master/results_ewnertc.csv

2.2 Approaches

In this section, we present the five models that we investigate for this study in more detail and we specify the configuration of each model.

2.2.1 CRF

As CRF models remain largely popular solutions for sequence-to-sequence tasks, we use a simple CRF model as a baseline. We use a large number of context and word shape features such as casing information and whether or not the word contains numerical characters. While simple CRF models generally perform well for coarse-grained NER, they require custom-made features and their usefulness is limited for FG-NER according to Mai et al. (2018) who observed that CRF models tend to require too much time to finish when handling a large number of labels. We use the sklearn_crfsuite API[6] for python with the following hyperparameters for training: gradient descent using the L-BFGS method as the training algorithm with a maximum of 100 iterations. The coefficients for L1 and L2 regularisation are fixed to $C_1 = 0.4$ and $C_2 = 0.0$. We use the following features: the word itself, casing information, is the word alphabetical, numerical or alphanumerical, suffixes and prefixes, as well as the words and features in a two-words context window. Considering that the datasets are numerous and very diverse, we decided against using specialised gazetteers/dictionaries for this study, despite their proven usefulness in earlier studies (Mai et al., 2018).

2.2.2 BiLSTM-CNN-CRF

As our state-of-the-art model, we use the implementation of Reimers and Gurevych (2017b)[7] of the BiLSTM-CNN-CRF model proposed by Ma and Hovy (2016). The model consists of a combination of a convolutional neural network (CNN) layer, a bidirectional long short-term memory (BiLSTM) layer, and a CRF layer. In a first step, the CNN is used to extract character-level representations of given words which are then concatenated with word embeddings to create word level representations of the input tokens. These representations are fed into a forward and a backward LSTM layer, creating a bidirectional encoding of the input sequence. Finally, a CRF layer decodes the resulting representations into the most probable label sequence (Ma and Hovy, 2016). Mai et al. (2018) achieved the best performance with a combination of gazetteers and BiLSTM+CNN+CRF, but as was mentioned above, we do not use gazetteers for this study due to the diverse nature of our datasets. We use the hyperparameters recommended by Reimers and Gurevych (2017a) as they were shown to be useful for coarse-grained NER. We also use **Glo**bal **Ve**ctors (GLoVe)[8] word embeddings with 300 dimensions for the same reason.

2.2.3 BERT

Pretraining a language model can take several days due to its large amount of trainable parameters. Furthermore, a sizable amount of data is required to achieve good results. Indeed, we tried to train a few language models using the EWNERTC dataset, but it is too small and the resulting models were essentially unusable as they yielded very low F1 scores. Fortunately, Google provides a variety of pretrained models that have been trained on the BooksCorpus (Zhu et al., 2015) and English Wikipedia, amounting to a grand total of 3.3 billion words. We use the Transformers library[9] provided by Huggingface (Wolf et al., 2019) which allows to pretrain and fine-tune BERT models with a simplified procedure using CLI commands. For this study, we fine-tune an English *BERT Base* model using each dataset separately. As we compare models for FG-NER, we chose the cased model as recommended, in order to preserve casing information. The *BERT Base* model contains 12 transformer blocks, 768 hidden layers, 12 self-attention blocks, and 110 million parameters in total. While the *BERT Large* model yields better results in every task that Devlin et al. (2019) investigated, the *BERT Base* model can be useful for determining a lower boundary for the performance. Devlin et al. (2019) report that the recommended hyperparameters vary depending on the NER task, but generally the best performances are observed for a batch size

[6] https://github.com/TeamHG-Memex/sklearn-crfsuite
[7] https://github.com/UKPLab/emnlp2017-bilstm-cnn-crf
[8] https://github.com/stanfordnlp/GloVe
[9] https://github.com/huggingface/transformers

in $\{16, 32\}$, a learning rate in $\{2^{-5}, 3^{-5}, 5^{-5}\}$, and training epochs in $\{2, 3, 4\}$. After testing on three specific domains (*comic books*, *symbols*, and *fictional universe* with 21 262, 21 171 and 39 781 sentences respectively), we found that a batch size of 16, a learning rate of 5^{-5}, and 5 training epochs yielded the highest F1 scores.

2.2.4 RoBERTa

RoBERTa presents similar challenges as BERT as it needs a large amount of resources, time and data. Liu et al. (2019) provide pretrained models, trained on 160GB of text, which represents about 3-4 times the amount of data used for pretraining BERT. We use the *RoBERTa Base* model, which contains 12 transformer blocks, 768 hidden layers, 12 self-attention heads, and 125 million trainable parameters. We fine-tune it on each dataset separately. Similar to the pretrained BERT model, the pretrained RoBERTa model is also cased, making it appropriate for fine-tuning on NER tasks. Liu et al. (2019) trained RoBERTa using the same hyperparameters as BERT, except for the number of training epochs which they fixed to ten. We perform a similar grid search as for BERT, i.e., a batch size in $\{16, 32\}$, and a learning rate in $\{2^{-5}, 3^{-5}, 5^{-5}\}$, but training epochs in $\{2, 4, 6, 8, 10\}$. Testing on the *comic books*, *symbols*, and *fictional universe*, we found that a batch size of 16 , a learning rate of 5^{-5}, and 10 training epochs performed best with regards to F1 score.

2.2.5 XLNet

While the pretraining approach of the XLNet model differs significantly from BERT models, the pretraining step still requires a vast amount of resources and time. Thus, we once again use a pretrained model rather than training one ourselves. For the comparison, we use the cased *XLNet Base* model with 12 transformer blocks, 768 hidden layers, 12 self-attention heads, and 110 million parameters. Yang et al. (2019) fine-tuned their pretrained model using the same hyperparameters as the BERT models to compare their performances. We perform the same hyperparameter grid search as for BERT, and get the best F1 score with a batch size of 16, a learning rate of 5^{-5} and 5 training epochs for the domains *comic books*, *symbols*, and *fictional universe*.

3 Experimental Results

In this section, we will answer the three research questions that we formulated for this study (cf. Section 1). Table 2 shows the performance of the five models for each domain. In order to account for the imbalanced distribution of the entity types, we opt to calculate micro-averaged performance scores which takes into account the frequency of every entity type. To facilitate reading, we highlight (in **bold**) the highest F1 score for each domain.

3.1 RQ1: Do transformer-based models outperform the state-of-the-art model for the FG-NER task?

The results indicate that, overall, the transformer-based models outperform CRF and BiLSTM-CNN-CRF in most domains in terms of F1 score. Specifically, the results show that the BERT and RoBERTa models yield the highest and second-highest F1 scores for almost every domain. BERT has the highest F1 score in 36 out of 49 domains, while RoBERTa achieves the best F1 score in 10 out of 49 domains. While XLNet outperforms BiLSTM-CNN-CRF in most domains, its performance scores are slightly lower than the ones of both the BERT and RoBERTa models. It is also noteworthy that XLNet performs consistently worse than BiLSTM-CNN-CRF in the ten smallest domains.

Figure 1a provides the boxplots showing the distributions of the F1 scores over all the domains across the five models. We can make two observations. The boxplots indicate that, on average, all of the transformer-based models achieve higher performances than both CRF and BiLSTM-CNN-CRF. Furthermore, we can observe that the ranges, and, more importantly, the interquartile ranges of the transformer-based models are smaller. This indicates that their performances are more stable and less sensitive to the choice of domain than the performances of CRF and BiLSTM-CNN-CRF.

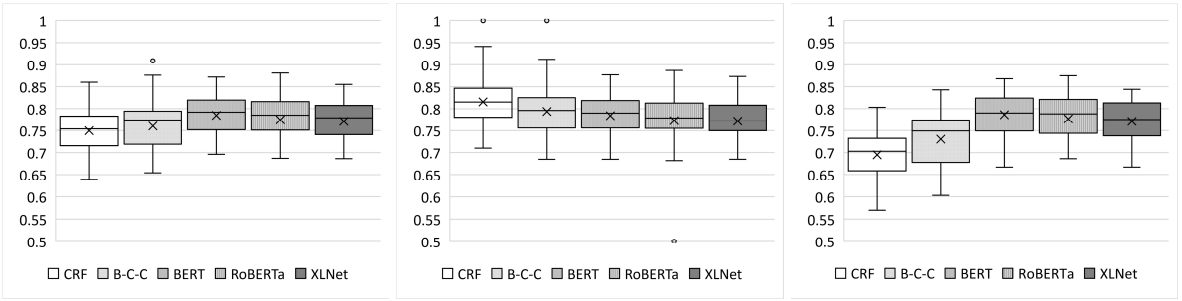
			CRF			BiLSTM-CNN-CRF			BERT			RoBERTa			XLNet		
ID	domain	#sentences	prec	rec	F1	prec	rec	F1	prec	rec	F1	prec	rec	F1	prec	rec	F1
1	physics	68	1	0.778	0.875	1	0.833	**0.909**	0.857	0.667	0.75	0.5	0.444	0.471	0.706	0.667	0.686
2	fashion	1043	0.92	0.765	**0.836**	0.894	0.776	0.831	0.849	0.801	0.824	0.816	0.816	0.816	0.825	0.77	0.797
3	finance	1723	0.859	0.708	0.776	0.83	0.731	0.777	0.807	0.796	0.802	0.794	0.839	**0.815**	0.768	0.759	0.764
4	exhibitions	1829	0.901	0.737	**0.811**	0.831	0.744	0.785	0.765	0.754	0.76	0.788	0.782	0.785	0.759	0.74	0.75
5	meteorology	2838	0.748	0.675	0.709	0.75	0.753	0.751	0.746	0.79	0.767	0.755	0.792	**0.773**	0.722	0.742	0.732
6	interests	3462	0.943	0.811	0.872	0.912	0.843	0.876	0.877	0.868	0.872	0.887	0.875	**0.881**	0.873	0.838	0.855
7	measurement unit	3864	0.822	0.707	0.76	0.812	0.772	0.791	0.794	0.806	**0.8**	0.79	0.795	0.792	0.773	0.785	0.779
8	internet	3915	0.83	0.63	0.716	0.768	0.657	0.709	0.727	0.712	0.719	0.749	0.725	**0.737**	0.73	0.687	0.708
9	engineering	4475	0.856	0.63	0.726	0.764	0.691	0.726	0.734	0.722	0.728	0.739	0.725	**0.732**	0.694	0.689	0.691
10	chemistry	4883	0.869	0.736	0.797	0.874	0.768	0.818	0.836	0.823	**0.829**	0.815	0.823	0.819	0.81	0.805	0.808
11	astronomy	8298	0.85	0.743	0.792	0.825	0.781	0.802	0.825	0.833	0.829	0.831	0.831	**0.831**	0.821	0.814	0.817
12	automotive	10 349	0.799	0.735	0.766	0.788	0.779	0.784	0.792	0.816	**0.803**	0.773	0.797	0.785	0.772	0.801	0.786
13	soccer	11 398	0.766	0.647	0.702	0.779	0.681	0.727	0.77	0.773	**0.772**	0.756	0.769	0.763	0.761	0.764	0.763
14	opera	11 559	0.865	0.74	0.798	0.825	0.776	0.8	0.827	0.847	**0.837**	0.83	0.839	0.834	0.814	0.824	0.819
15	law	11 813	0.792	0.64	0.708	0.756	0.701	0.727	0.75	0.759	0.754	0.758	0.752	**0.755**	0.761	0.745	0.753
16	visual art	12 059	0.861	0.649	0.74	0.81	0.674	0.736	0.766	0.725	0.745	0.774	0.721	**0.747**	0.761	0.718	0.738
17	basketball	12 604	0.836	0.796	0.815	0.832	0.83	0.831	0.833	0.849	**0.841**	0.828	0.85	0.839	0.824	0.844	0.834
18	computer	12 955	0.814	0.673	0.737	0.768	0.74	0.754	0.762	0.773	**0.767**	0.755	0.767	0.761	0.748	0.757	0.752
19	theater	15 340	0.79	0.608	0.688	0.733	0.658	0.694	0.709	0.719	0.714	0.719	0.725	**0.722**	0.7	0.697	0.698
20	symbols	21 171	0.72	0.571	0.637	0.715	0.62	0.664	0.723	0.727	**0.725**	0.724	0.712	0.718	0.711	0.699	0.705
21	comic books	21 262	0.854	0.711	0.776	0.808	0.749	0.777	0.808	0.829	0.818	0.818	0.821	**0.82**	0.796	0.815	0.805
22	language	21 306	0.803	0.74	0.77	0.79	0.764	0.777	0.81	0.816	**0.813**	0.799	0.809	0.804	0.787	0.8	0.793
23	religion	27 977	0.805	0.697	0.747	0.787	0.761	0.774	0.808	0.81	**0.809**	0.8	0.796	0.798	0.787	0.791	0.789
24	time	28 903	0.717	0.565	0.632	0.697	0.63	0.662	0.716	0.722	**0.719**	0.704	0.704	0.704	0.704	0.705	0.705
25	royalty	30 587	0.804	0.725	0.762	0.785	0.76	0.772	0.786	0.798	**0.792**	0.779	0.788	0.784	0.774	0.785	0.779
26	games	31 420	0.839	0.741	0.787	0.796	0.77	0.783	0.79	0.813	**0.801**	0.789	0.81	0.799	0.768	0.791	0.779
27	aviation	36 924	0.795	0.712	0.751	0.779	0.73	0.754	0.789	0.807	**0.798**	0.781	0.797	0.789	0.774	0.79	0.782
28	medicine	37 729	0.848	0.697	0.765	0.797	0.755	0.776	0.802	0.788	**0.795**	0.791	0.788	0.789	0.799	0.791	**0.795**
29	fictional universe	39 781	0.874	0.756	0.811	0.845	0.781	0.812	0.843	0.855	**0.849**	0.841	0.848	0.845	0.837	0.842	0.839
30	food	41 160	0.801	0.648	0.717	0.746	0.69	0.717	0.776	0.788	**0.782**	0.76	0.766	0.763	0.752	0.774	0.763
31	media common	49 714	0.862	0.723	0.786	0.819	0.755	0.786	0.806	0.825	**0.815**	0.807	0.819	0.813	0.803	0.812	0.807
32	biology	53 042	0.854	0.771	0.811	0.843	0.807	0.825	0.834	0.847	**0.84**	0.832	0.837	0.834	0.836	0.837	0.836
33	travel	59 965	0.822	0.696	0.754	0.803	0.719	0.759	0.784	0.79	**0.787**	0.764	0.772	0.768	0.779	0.777	0.778
34	business	68 244	0.803	0.634	0.709	0.756	0.666	0.708	0.765	0.771	**0.768**	0.755	0.759	0.757	0.752	0.754	0.753
35	architecture	76 322	0.709	0.588	0.643	0.685	0.627	0.654	0.707	0.722	**0.715**	0.688	0.701	0.694	0.685	0.695	0.69
36	geography	94 712	0.813	0.728	0.768	0.801	0.752	0.776	0.798	0.815	**0.806**	0.795	0.804	0.799	0.789	0.799	0.794
37	military	95 809	0.836	0.731	0.78	0.82	0.778	0.798	0.816	0.827	**0.821**	0.811	0.821	0.816	0.809	0.823	0.816
38	transportation	111 864	0.828	0.738	0.781	0.834	0.804	0.819	0.845	0.857	**0.851**	0.845	0.85	0.848	0.839	0.844	0.841
39	award	117 280	0.702	0.617	0.657	0.702	0.671	0.686	0.685	0.716	**0.7**	0.682	0.707	0.694	0.689	0.703	0.695
40	book	135 865	0.761	0.604	0.675	0.717	0.639	0.676	0.711	0.73	**0.721**	0.708	0.723	0.716	0.716	0.722	0.719
41	organization	146 583	0.769	0.64	0.698	0.765	0.674	0.717	0.767	0.776	**0.771**	0.756	0.766	0.761	0.762	0.768	0.765
42	tv	154 152	0.725	0.574	0.641	0.733	0.603	0.662	0.697	0.696	**0.696**	0.688	0.686	0.687	0.702	0.684	0.693
43	sports	171 645	0.781	0.705	0.741	0.799	0.767	0.783	0.806	0.822	**0.814**	0.801	0.816	0.808	0.807	0.819	0.813
44	education	212 423	0.734	0.653	0.691	0.747	0.706	0.726	0.769	0.78	**0.774**	0.763	0.774	0.769	0.769	0.774	0.771
45	government	331 720	0.81	0.725	0.765	0.815	0.764	0.789	0.821	0.828	**0.825**	0.816	0.824	0.82	0.824	0.825	0.824
46	film	478 479	0.75	0.68	0.713	0.743	0.695	0.718	0.769	0.773	**0.771**	0.766	0.767	0.766	0.772	0.768	0.77
47	music	462 949	0.786	0.654	0.714	0.78	0.668	0.72	0.744	0.744	**0.744**	0.739	0.736	0.737	0.752	0.736	**0.744**
48	people	442 683	0.836	0.771	0.802	0.847	0.795	0.82	0.83	0.83	**0.83**	0.825	0.821	0.823	0.834	0.825	0.829
49	location	443 646	0.809	0.703	0.752	0.8	0.713	0.754	0.79	0.789	**0.79**	0.775	0.772	0.774	0.784	0.775	0.78

Table 2: Micro-averaged results of each model for every domain. Bold text indicates the highest F1 score for the domain.

3.2 RQ2: What are the strengths, weaknesses, and trade-offs of each investigated model?

While the transformer-based models clearly outperform the other models with regards to the F1 score, it is worth examining the precision and recall scores as well. Regarding the precision, the CRF model almost consistently outperforms all of the other models as shown in Table 2. When compared to the BiLSTM-CNN-CRF model, the transformer-based models perform worse in most domains in terms of precision. In fact, BERT outperforms BiLSTM-CNN-CRF in less than half of the domains, RoBERTa outperforms BiLSTM-CNN-CRF in only a third of the domains and XLNet outperforms it in only a fifth of the domains. Figure 1b shows the distribution of the precision scores over all the domains across the five models. The boxplots confirm the strength of CRF over the other models. Furthermore, they show that BiLSTM-CNN-CRF performs slightly better than the transformer-based models, albeit at a loss of stability as indicated by the large range.

On the other hand, the transformer-based models significantly outperform the other models with regards to recall as seen in Table 2. In fact, both BERT and RoBERTa significantly outperform CRF and BiLSTM-CNN-CRF in almost every domain, while XLNet outperforms them in most. The same result



(a) Distribution of F1 scores (b) Distribution of Precision (c) Distribution of Recall

Figure 1: Distribution the performance of the five models used

can be observed in Figure 1c. The transformer-based models not only outperform the other models, but their interquartile ranges are significantly smaller as well. This difference in recall score also explains the higher F1 scores for the transformer-based models.

To summarise, CRF shows its strength in terms of precision, BERT, RoBERTa, and XLNet perform well with regards to both recall and F1 score, with BERT usually achieving the highest performances. The BiLSTM-CNN-CRF model acts as a trade-off between CRF and the transformer-based models.

3.3 RQ3: How does the choice of the domain influence the performance of the models?

Figure 1a shows that while different models may achieve significantly different performance, no approach yields a significant breakthrough, w.r.t the others, for the task at hand, and all leave room for improvement. The five tested models obtained relatively stable performances, as is visible from the fact that boxes, which represent the performance measurements of 50% of the domains, cover only a ± 0.05 band around the average.

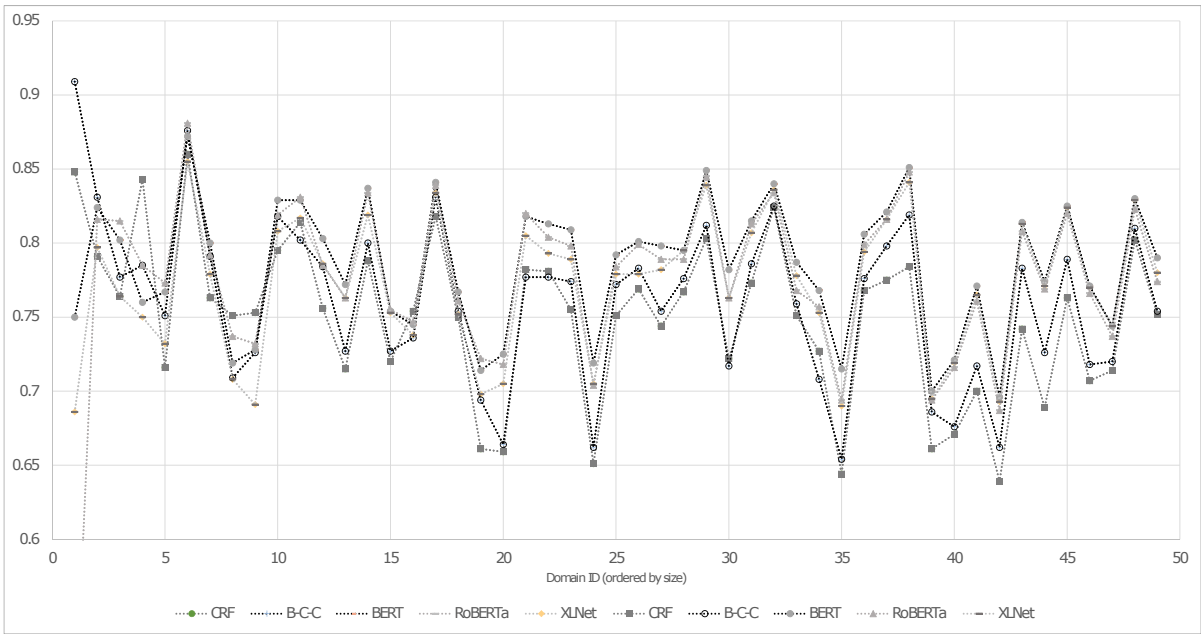


Figure 2: Performances of the five models for every domain.

Figure 2, that plots the F1 scores for every domain (ordered by size), reveals however that all models are similarly impacted by domains: with the exceptions of the four smallest domains (left-most on Figure 2), when one model achieves a lower performance than its overall average, all models are also performing worse than their overall averages. We also note that the per-domain variations in performance cannot be explained by the size of the domains (since the performance looks erratic across all domain

sizes). Overall, the results are a clear indication that most domains are either: (a) *relatively hard* for every model, or (b) *relatively easy* for every model. This suggests that no model manages to acquire a massively better language *understanding* that would make it able to avoid the difficulties faced by the other models, at least in the context of FG-NER.

Furthermore, the ranking of the five models is very stable across domains: given the fact that one specific model performs the best (resp. the worst) for one domain, it can reliably be predicted that this model will also perform the best (resp. the worst) across all domains. It follows that some models do bring a sometime incremental, but nonetheless measurable improvement over other models. Nevertheless, we note that for the four smallest domains, the difference in performance from one model to another is more important, and no ranking pattern is visible.

The performance variations between domains that we see in our results have also been reported in the study by Guo et al. (2006), who investigated the stability of coarse-grained NER across domains for the Chinese language. Notably, when trained on the *sports* domain, their baseline has a significantly higher F1-score than the other domains. The same is true here, but it has to be noted that they use the classic NER-labels, i.e., *person*, *location*, *organisation*, and *miscellaneous*, rather than domain-specific labels.

Take-Home Messages: To summarise, the transformer-based models do indeed outperform the BiLSTM-CNN-CRF model with regards to F1 score, with BERT yielding the highest results overall. The simple CRF model achieved the best performance in terms of precision, while performing the worst in terms of recall. Compared to both CRF and BiLSTM-CNN-CRF, the transformer-based models achieved significantly higher recall scores. Furthermore, we observe significant discrepancies when applying the models to different domains. Moreover, when a model is performing better (resp. worse) on one domain, the other models also perform better (resp. worse). This suggests that while transformer-based models can indeed bring significant performance improvements, their language *understanding* may not be outstandingly different. Indeed, if they were clearly different, we could have reasonably expected to note different patterns in the performance for the FG-NER task (i.e., they would not systematically perform well/badly for the same domains).

4 Related Work

4.1 Fine-Grained Named Entity Recognition

Early efforts to develop a fine-grained approach to NER were made by Béchet et al. (2000), where they focused on differentiating between first names, last names, countries, towns, and organisations. While this would be considered coarse-grained by today's standards, they do split the classical NER labels *person* and *location* into more nuanced labels. FG-NER was first described as "fine grained classification of named entities" by Fleischman and Hovy (2002). They focused on a fine-grained label set for personal names, dividing the generic *person* label into eight subcategories, i.e., *athlete*, *politician/government*, *clergy*, *businessperson*, *entertainer/artist*, *lawyer*, *doctor/scientist*, and *police*. They experimented with a variety of classic machine learning approaches for this task, and achieved promising results of 68.1%, 69.5%, and 70.4% in terms of accuracy for SVM, a feed-forward neural network, and a C4.5 decision tree, respectively. Furthermore, Ling and Weld (2012) introduced their fine-grained entity recognizer (FIGER), which can distinguish between 112 different labels and handle multi-label classification.

Mai et al. (2018) presented an empirical study on FG-NER prior to the rise of transformer-based models (which are the focus of our study). They targeted an English dataset containing 19 800 sentences and a Japanese dataset which contained 19 594 sentences, dividing the named entities into 200 categories. They compared performances for FIGER, BiLSTM-CNN-CRF, and a hierarchical CRF+SVM classifier, which classifies an entity into a coarse-grained category before further classifying it into a fine-grained subcategory. Furthermore, they combine some of the aforementioned methods with gazetteers and category embeddings to further improve the performance of the models. They found that the BiLSTM-CNN-CRF model by Ma and Hovy (2016) combined with gazetteer information performed the best for the English language with an F1 score of 83.14% while BiLSTM-CNN-CRF with both gazetteers and

category embeddings yielded an F1 score of 82.29%, and 80.93% without either gazetteers or category embeddings.

4.2 The Rise of Transformers

Vaswani et al. (2017) first described the transformer model which superseded the popular LSTM model in favour of the attention mechanism (Bahdanau et al., 2014). As transformers do not need to process sentences in sequence, they allow for more parallelisation than LSTMs or other recurrent neural network models. Due to this advantage, transformers have become fundamental for state-of-the-art models in the NLP field. One early notable model that employed transformers is the Generative Pretraining Transformer (GPT) model (Radford et al., 2018) which outperformed state-of-the-art models in nine out of twelve NLU tasks. Devlin et al. (2019) further revolutionised the NLP landscape by introducing BERT. Unlike the unidirectional GPT model, BERT is a deeply bidirectional transformer model, pretrained on the MLM and NSP tasks. Fine-tuned BERT models managed to outperform state-of-the-art models in eleven NLP tasks, including the GLUE (Wang et al., 2018) and SQuAD (Rajpurkar et al., 2016) benchmarks. The success of BERT led to a large variety of similar models, which were pretrained on different datasets. Most notably, RoBERTa (Liu et al., 2019) and XLNet managed to further outperform BERT in a large number of tasks. Specifically, Yang et al. (2019) introduced XLNet, replacing the MLM task with a permutation-based autoregression task, effectively predicting sentence tokens in random order. XLNet manages to outperform BERT in 20 tasks, including the GLUE, SQuAD and RACE (Lai et al., 2017) benchmarks. Meanwhile, the RoBERTa model was trained on more data, for longer periods of time, tweaked the MLM pretraining task, and removed the NSP task. Liu et al. (2019) reported that RoBERTa outperforms BERT on the GLUE, SQuAD, and RACE benchmarks.

5 Threats to Validity

This study was conducted on the EWNERTC dataset (Sahin et al., 2017a) which was annotated automatically. We are operating under the assumption that the annotations are accurate. However, while Sahin et al. (2017b) conducted an evaluation for the Turkish counterpart of the dataset (TWNERTC), they did not evaluate the English one. Nevertheless, EWNERTC is the largest publicly available dataset that we could find and that is relevant for FG-NER studies. We further proposed to reduce the potential noise in labelling by considering only the subset associated to top labels (cf. Section 2.1).

Performance measurements can be impacted by sub-optimal implementation of algorithms. To mitigate this threat, we collected the models' implementations that were released by their original authors, and already leveraged in previous studies, and we reused them in the settings they were designed for.

While we conducted grid searches to determine optimised hyperparameters for the CRF, BERT, RoBERTa and XLNet models, we did not specifically optimise the hyperparameters for the the BiLSTM-CNN-CRF model due to the induced computational costs. Furthermore, as pointed out in section 2, due to the large number of domains, we decided against using gazetteers even though they would likely have increased the F1-scores of the non-transformer-based models.

6 Conclusion

In this paper, we presented an empirical study of the performance of various transformer-based models for the FG-NER task on a multitude of domains and compared them to both CRF and BiLSTM-CNN-CRF models (which are commonly used in the literature for the NER task).

We concluded that while the transformer-based models did not manage to outperform non-transformer-based models in terms of precision, we observed a consistent increase in recall and F1 scores in most domains. We noticed, however, significant differences in performance for a selection of domains that could not be explained by the size of the respective datasets. This study yields the main insight that while transformer-based models can indeed bring significant performance improvements, they do not necessarily revolutionise the achievements in FG-NER to the same extent they did in other NLP tasks.

References

Alan Akbik, Duncan Blythe, and Roland Vollgraf. 2018. Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Frédéric Béchet, Alexis Nasr, and Franck Genet. 2000. Tagging unknown proper names using decision trees. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 77–84. 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*, pages 4171–4186.

Michael Fleischman and Eduard Hovy. 2002. Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics.

Hong Lei Guo, Li Zhang, and Zhong Su. 2006. Empirical study on the performance stability of named entity recognition model across domains. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 509–516.

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.

Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. 2017. Race: Large-scale reading comprehension dataset from examinations. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 785–794.

Xiao Ling and Daniel S. Weld. 2012. Fine-grained entity recognition. In *Proceedings of the Twenty-Sixth AAAI Conference on Artificial Intelligence*, AAAI'12, page 94–100. AAAI Press.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

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*, pages 1064–1074, Berlin, Germany, August. Association for Computational Linguistics.

Khai Mai, Thai-Hoang Pham, Minh Trung Nguyen, Tuan Duc Nguyen, Danushka Bollegala, Ryohei Sasano, and Satoshi Sekine. 2018. An empirical study on fine-grained named entity recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 711–722, Santa Fe, New Mexico, USA, August. Association for Computational Linguistics.

Matthew Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. 2017. Semi-supervised sequence tagging with bidirectional language models. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1756–1765.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training. `http://openai-assets.s3.amazonaws.com/research-covers/language-unsupervised/language_understanding_paper.pdf`.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392.

Nils Reimers and Iryna Gurevych. 2017a. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *arXiv preprint arXiv:1707.06799*.

Nils Reimers and Iryna Gurevych. 2017b. 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 (EMNLP)*, pages 338–348, Copenhagen, Denmark, 09.

H. Bahadir Sahin, Mustafa Tolga Eren, Caglar Tirkaz, Ozan Sonmez, and Eray Yildiz. 2017a. English/turkish wikipedia named-entity recognition and text categorization dataset.

H Bahadir Sahin, Caglar Tirkaz, Eray Yildiz, Mustafa Tolga Eren, and Ozan Sonmez. 2017b. Automatically annotated turkish corpus for named entity recognition and text categorization using large-scale gazetteers. *arXiv preprint arXiv:1702.02363*.

Erik 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.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355.

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

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5754–5764.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium, October-November. Association for Computational Linguistics.

Yukun Zhu, Ryan Kiros, Rich Zemel, Ruslan Salakhutdinov, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Aligning books and movies: Towards story-like visual explanations by watching movies and reading books. In *Proceedings of the IEEE international conference on computer vision*, pages 19–27.