Bridging the Gap: A Comprehensive Study on Named Entity Recognition in Electronic Domain using Hybrid Statistical and Deep Knowledge Transfer

Ghaith Dekhili, Ngoc Tan Le, Fatiha Sadat
Université du Québec à Montréal, Canada H2X 3Y7
dekhili.ghaith@courrier.uqam.ca, le.ngoc_tan@uqam.ca, sadat.fatiha@uqam.ca

Abstract

Training deep neural network models in NLP applications with a small amount of annotated data does not usually achieve high performances. To address this issue, transfer learning, which consists of transferring knowledge from a domain with a large amount of annotated data to a specific domain which lacks annotated data, could be a solution. In this paper, we present a study case on named entity recognition for the electronic domain, that relies on several approaches based on statistics, deep learning, and transfer learning. Our evaluations showed a significant improvement in overall performance, with the best results using transfer learning, up to $+15\%$ compared to other approaches. As Transformers-based models have shown their effectiveness in many NLP tasks in the last years, in this study, we compare our models performance to some Transformers-based models.

Introduction

Named Entity Recognition (NER) consists of a subtask of Natural Language Processing (NLP) and Information Extraction (IE). It involves the identification of specific textual elements, such as the names of people, organizations, and locations (Yadav and Bethard 2018). Transfer learning, particularly through the use of pre-trained language models that are specific to a domain, has demonstrated encouraging results by taking advantage of large-scale language models that have been pre-trained on a variety of text corpora and then fine-tuning them on data that are specific to a domain (Pan and Yang 2010; Meftah, Semmar, and Sadat 2018).

In the first step of our work, we present a study case in two approaches, statistical approach and neural network based approach, to deal with the NER task. In the second step, we propose a transfer learning-based method for NER from a general domain towards a specific domain, which is the electronic domain.

Our contribution in this study is twofold. First, we propose the design of a NER system for a specific low-resource domain; which is characterized by the challenge of lack of sufficient amount of labeled data, in addition to a high number of named entity classes, compared to the state of the art. Second, we proceed by transferring knowledge from a source domain to a target domain, the electronic domain, using different labels or classes.

To the best of our knowledge, the majority of existing methods are designed for image and text classification problems rather than sequence classification problems. Also, most existing methods of the state-of-the-art have been developed under the assumption that the source and target domains have the same class labels (Day and Khoshgoftaar 2017). Furthermore, many corpora used in the NER task have only a small number of annotated categories. For these reasons, the problem presented in this study could be considered more challenging.

This paper is presented as follows: Section Related Work presents relevant previous works. Section Methodology presents our proposed approaches. Section Experiments and Evaluations presents used tools and datasets, along with the results of our experiments. Section Conclusion gives some conclusions and ideas for future work.

Related Work

Named Entity Recognition is a key task in NLP and IE. It focuses on identifying and categorizing named entities within text, using several main approaches, such as:

Rule-based approaches: These methods rely on predefined rules, patterns, or dictionaries to identify entities (Nadeau and Sekine 2007). Rules can be based on linguistic patterns, regular expressions, or specific formatting within the text. While they can be precise, they might lack adaptability to new or varied data (Quimbaya et al. 2016).

Statistical and Machine Learning approaches:

- Supervised learning: Uses labeled data to train a model to recognize entities based on features derived from text (*for example*, word embeddings, part-of-speech tags). Models like Conditional Random Fields (Lafferty, McCallum, and Pereira 2001; Joshi et al. 2015), Support Vector Machines, and neural networks (*for example*, BiLSTMs, Transformers) are commonly used (Lample et al. 2016).

- Semi-supervised learning: Utilizes a combination of labeled and unlabeled data. Techniques like self-training or co-training help improve models' performance by leveraging both types of data (Zafarian et al. 2015; Chen et al. 2023).

- Unsupervised learning: Focuses on identifying patterns in text without labeled data. Clustering, topic modeling, or pattern discovery methods can be used, although these often lack precision compared to supervised approaches (Collobert et al. 2011).

Deep learning approaches: Neural network architectures have shown significant advances in NER. Models like Bidirectional Encoder Representations from Transformers (BERT) (Devlin et al. 2018), GPT (Generative Pre-trained Transformer) (Brown et al. 2020), or its variants are employed to capture contextual information effectively. These models leverage large-scale pretraining on vast amounts of text to learn rich representations, which can then be fine-tuned on specific NER tasks.

Hybrid approaches: Combining multiple techniques to capitalize on their strengths and mitigate weaknesses. For instance, a hybrid model might use rule-based systems to handle specific cases where patterns are well-defined and then use machine learning or deep learning models for more general cases (Haider et al. 2023; Wang et al. 2023; Bharathi et al. 2024).

Transfer learning approach: This approach consists of transferring knowledge from general NER models to specific domains such as healthcare, biomedical domain (Sachan et al. 2018), finance (Zhang et al. 2023), or legal documents (Çetindağ, Yazıcıoğlu, and Koç 2023; Naik, Patel, and Kannan 2023), by using domain-specific data or by fine-tuning existing models on domain-specific datasets (Sabane et al. 2023). This approach helps improve the accuracy and relevance of entity recognition within a particular field.

Methodology

This study aims to compare the role of transfer learning in improving performances of statistical-based model, statistical-based model augmented with orthographic features and neural network-based model.

Statistical-based approach

Basic features We proceeded by converting the training datasets into 4 columns: tokens, POSTags, ChunkTags and classes of entity. These columns represent basic features to train our statistical CRF model.

Additional orthographic features We added orthographic features extracted from words. Along with these features, we quote N-grams. Then we check if a word is capitalized, if it is a title, or if it is a special character.

Neural network approach

Inspired of (Lample et al. 2016), we used, in our architecture, BiLSTM layers for word-level features extraction. We used a CRF layer on the top, augmented with some features such as dropout layers.

Apart from using word representations, we used also character representations to capture morphological and orthographic information.

Input Embeddings The input layers of our model are vector representations of individual words. In our study, we used pre-trained word embeddings to initialize our look-up table and to enrich our training dataset. Learning character-level embeddings has the advantage of learning representations specific to the task and domain at hand. Following (Chiu and Nichols 2015), we used a CNN layer to derive a representation of each word from its characters.

Additional features As information related to capitalization has been deleted during word embeddings' map construction, we used a separate look-up table to add this feature (Collobert et al. 2011; Chiu and Nichols 2015). We also used additional character-based features by using a look-up table which generates a 4 dimensional vector representing the character's type (uppercase, lowercase, punctuation, and other).

Transfer Learning-based approach

In our work, we investigated the idea of transferring knowledge from general domain with enough amount of labeled data to a specific domain with a small amount of labeled data to train a deep neural network model.

First, we trained a model from scratch on CoNLL03 dataset. Second, we performed a transferring of pre-trained weights to initialize weights of the target model. These weights are then fine-tuned when training the target model on the target dataset. Both models have the same architecture and different layers have same dimensions, except the last CRF layer whose dimension depends on entities number. In the next section, we evaluated the performance of our transfer learning-based method to other state-of-the-art Transformer-based models.

Experiments and Evaluations

Data preparation

General domain data Many works on NER report their models performance in CoNLL03 (Table 1), which is one of the most used datasets to compare the performance of different existing models. The corpus contains four types of named entities: location (LOC), person (PER), organization (ORG) and miscellaneous (MISC).

Datasets	#tokens	#sentences
Train	204,562	14,985
Dev	51,573	3,464
Test	46,624	3,682
Total	302,759	22,131

Table 1: Statistics of the CoNLL03 dataset

Specific domain data The specific domain data had been extracted from the Comcast XFINITY forum as raw data and annotated by domain experts. The corpus contains 33 types of entities. The statistics of this corpus are shown in (Table 2).

Training settings

Our experiments used the IOB tagging schema, Inside, Outside, and Begin, which marks the token's position in the named entity (Chiu and Nichols 2015). Except for character

Datasets	#tokens	#sentences
Train	85,000	5,080
Dev	17,000	1,016
Test	11,330	677
Total	113,300	6,774

Table 2: Statistics of specific domain dataset

and word embeddings whose initializations were described previously, all lookup tables are randomly initialized with values drawn from the standard normal distribution. Training is done by using mini-batch with *nadam* optimization algorithm with a fixed learning rate. Our model used a single layer for the forward and backward LSTMs. To encourage the dependence of the model on word and character representations, we used dropout layers and set the rate at 0.5.

We observed a significant improvement in the performance of our model after dropping out. Applying dropout to the output nodes of each LSTM layer was quite effective in reducing overfitting (Lample et al. 2016).

Evaluations

	Model 1	Model 2	Model 3	Model 4 (ours)
P	0.730	0.794	0.743	0.824
R	0.568	0.661	0.704	0.727
F1	0.622	0.712	0.723	0.773

Table 3: Performance of all models, evaluated on the specific domain dataset, using the statistical model with only basic features (Model 1), the statistical model with basic + orthographic features (Model 2), the neural model without applying knowledge transfer (Model 3) and the neural model with knowledge transfer (Model 4, our model).

Statistical-based approach

According to the first and second columns of Table 3 (Model 1 and Model 2, respectively), we noticed that the model with orthographic features leads to a notable increase in precision and recall averages, which obtained an improvement of approximately $+9\%$ in terms of F1 compared to the statistically based model. All these improvements justify the important role of orthographic information in NER task.

Neural network-based approach

Between the model 1 and the model 3, we noticed a significant improvement of our neural approach, in the model 4, with 0.773 against 0.622 and 0.723 in terms of F1, respectively (Table 3).

All these performance increases proved the efficiency of our neural model architecture and also from multiple features during the training, especially pre-trained word embeddings trained on large quantities of raw data and character embeddings extracted from specific domain dataset. Therefore, our neural model is able to extract relevant knowledge from training data, without using hand-made features.

Transfer Learning approach

In this section, we start with presenting the mapping we have done to transfer knowledge from CoNLL03 data to specific domain data.

Mapping: In order to perform the correspondence between the indexing of the source and target datasets, we defined a mapping between different types of named entity. This mapping is a sort of equivalence between entities in the two datasets (Table 4).

CoNLL dataset		Specific domain dataset	
Entity	**Index**	**Entity**	**Index**
O	0	O	0
B-LOC	1	B-Location	1
B-PER	2	B-User_Name	2
B-ORG	3	B-TV_Manufacturer	3
I-PER	4	I-User_Name	4
I-ORG	5	I-TV_Manufacturer	5
I-LOC	6	I-Location	6
B-MISC	7	B-MISC	7
I-MISC	8	I-MISC	8
		...	...

Table 4: Mapping between source and target entities

We noticed that the knowledge transfer from source data to target data improves averages of precision, recall, and F1 of all evaluated entities. The application of transfer learning showed a significant improvement in terms of F1 (Model 4), with $+15\%$ compared to the basic statistical model, $+6\%$ compared to the statistical model enriched with orthographic features, and $+5\%$ compared to the basic neural model (Table 3).

Transformer-based models

In this section, we evaluated the transfer learning technique used in some Transformer-based models, such as BERT-base (Devlin et al. 2018), XLNet (Yang et al. 2019), RoBERTa (Liu et al. 2019), and DistilBERT (Sanh et al. 2019), by fine-tuning them on our specific domain dataset for the NER task (Table 5).

	BERT-base	XLNet	RoBERTa	DistilBERT
P	0.708	0.682	0.722	0.714
R	0.782	0.769	0.779	0.761
F1	0.743	0.723	0.750	0.737

Table 5: Performance of all Transformer-based models such as BERT-base, XLNet, RoBERTa, and DistilBERT.

We observed that the RoBERTa model performed better on average compared to other models. This can be explained by the fact that the BERT-based models have been pre-trained on far more data (over 142 GB) compared to BERT and DistilBERT (with only 13 GB).

Furthermore, in RoBERTA and XLNet, Liu et al. (2019) and Yang et al. (2019) used a more efficient technique that leads to better performance in many NLP tasks. We also found that the DistilBERT model performed well enough, even if it is a lighter and less costly model.

When we compared the performance between our models and Transformer-based models (Tables 3 and 5), we noticed that our approach (Model 4), in which we transfer knowledge from the general domain dataset to the specific domain dataset, outperformed on average against the best of Transformer-based models, the RoBERTa model, with 0.773 against 0.750 in terms of F1, respectively, which is an increase of $+2.3\%$.

The main reason for this can be attributed to the fact that we explicitly transfer knowledge for some entities by aligning them with other entities such as *LOC* and *ORG* in the source dataset. This alignment allows the target model to leverage more relevant knowledge from the source domain. This shows the effectiveness of transferring knowledge explicitly using a high-quality labeled dataset of the same task, compared to fine-tuning language models pretrained on general unlabeled data.

If we compare the performance of the models on entities taken separately, we notice that our model outperforms XLNet in terms of precision, recall, and F1, on *Location* and *TV_Manufacturer* entities (Tables 6 and 7). This can be explained mainly by the fact that we transfer knowledge for these entities explicitly by aligning them with *LOC* and *ORG* entities in the source dataset, which gave the target model the ability to take advantage of the knowledge in the source domain. We notice that our model performs better on *TV_Box_Device* and *Networking_Feature* as well. It should be noted that model 1 and model 3 perform better on *Software_Device*, *Network_Service_Provider* and model 3 performs better on *video_feature* and *Networking_Manufacturer* compared to Transformer-based models. Despite the fact that Transformer-based models perform worse on the aforementioned entities, they slightly outperform our models on other entities which are *Service_Offering*, *User_Name*, *Model_Name*, *Software_Manufacturer*, *version*, *Networking_Protocol* and *TV_Box_Manufacturer*. This can be explained by the fact that even if the dataset on which these models have been fine-tuned is small, they still manage to recognize entities classes by leveraging knowledge from then huge amount of data on which they have been pretrained.

Another potential explanation could be attributed to the fact that, despite the small size of the dataset used for finetuning these models, they are still identifying different entity classes by utilizing the extensive knowledge gained from the large amount of pre-training data.

Conclusion

In this study, we presented the NER task for the electronic domain by applying multiple methods such as statistical-based, neural network-based, and transfer deep learning-based methods.

In our experiments, we used features related to words syntax. Besides we used word and character embeddings, which allows us to detect morphological and orthographic information and learn task specific representations. According to our results, we conclude that transfer learning gives best results and increases up to $+15\%$ in terms of F1 compared to state-of-the-art models.

To the best of our knowledge, there is no study relating the application of transfer learning from general domain to electronic domain in NER with much more entity classes and different from those of the source domain. In future work, we plan to use other features in the hybrid architecture and investigate external resources such as ontologies and knowledge bases.

References

Bharathi, A.; Ramdin, R.; Babu, P.; Menon, V. K.; Jayaramakrishnan, C.; and Lakshmikumar, S. 2024. A hybrid named entity recognition system for aviation text. *EAI Endorsed Transactions on Scalable Information Systems* 11(1).

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language models are few-shot learners. *CoRR* abs/2005.14165.

Çetindağ, C.; Yazıcıoğlu, B.; and Koç, A. 2023. Named-entity recognition in turkish legal texts. *Natural Language Engineering* 29(3):615–642.

Chen, Y.; Wu, C.; Qi, T.; Yuan, Z.; Zhang, Y.; Yang, S.; Guan, J.; Sun, D.; and Huang, Y. 2023. Semi-supervised named entity recognition in multi-level contexts. *Neurocomputing* 520:194–204.

Chiu, J. P. C., and Nichols, E. 2015. Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics* 4(1):357–370.

Collobert, R.; Weston, J.; Bottou, L.; Karlen, M.; Kavukcuoglu, K.; and Kuksa, P. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12:2493–2537.

Day, O., and Khoshgoftaar, T. M. 2017. A survey on heterogeneous transfer learning. *Journal of Big Data* 4:1–42.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR* abs/1810.04805.

Haider, A.; Saeed, S.; Bilal, K.; and Erbad, A. 2023. A hybrid approach for food name recognition in restaurant reviews. In *2023 International Symposium on Networks, Computers and Communications (ISNCC)*, 1–6. IEEE.

Joshi, M.; Hart, E.; Vogel, M.; and Ruvini, J.-D. 2015. Distributed word representations improve NER for e-commerce. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 160–167. Colorado: Association for Computational Linguistics.

Lafferty, J. D.; McCallum, A.; and Pereira, F. C. N. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, 282–289. Morgan Kaufmann Publishers Inc.

Lample, G.; Ballesteros, M.; Subramanian, S.; Kawakami, K.; and Dyer, C. 2016. Neural architectures for named entity recognition. In *Proceedings of NAACL-HLT 2016*, 260–270.

Liu, Y.; Ott, M.; Goyal, N.; Du, J.; Joshi, M.; Chen, D.; Levy, O.; Lewis, M.; Zettlemoyer, L.; and Stoyanov, V. 2019. Roberta: A robustly optimized bert pretraining approach.

Meftah, S.; Semmar, N.; and Sadat, F. 2018. A Neural Network Model for Part-Of-Speech Tagging of Social Media Texts. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2821 – 2828. Japan: European Language Resources Association (ELRA).

Nadeau, D., and Sekine, S. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes* 30(1):3–26.

Naik, V.; Patel, P.; and Kannan, R. 2023. Legal entity extraction: An experimental study of ner approach for legal documents. *International Journal of Advanced Computer Science and Applications* 14(3).

Pan, S. J., and Yang, Q. 2010. A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* 22(10):1345–1359.

Quimbaya, A. P.; Múnera, A. S.; Rivera, R. A. G.; Rodríguez, J. C. D.; Velandia, O. M. M.; Peña, A. A. G.; and Labbé, C. 2016. Named entity recognition over electronic health records through a combined dictionary-based approach. *Procedia Computer Science* 100(1):55 – 61.

Sabane, M.; Ranade, A.; Litake, O.; Patil, P.; Joshi, R.; and Kadam, D. 2023. Enhancing low resource ner using assisting language and transfer learning. In *2023 2nd International Conference on Applied Artificial Intelligence and Computing (ICAAIC)*, 1666–1671. IEEE.

Sachan, D. S.; Xie, P.; Sachan, M.; and Xing, E. P. 2018. Effective use of bidirectional language modeling for transfer learning in biomedical named entity recognition. In *Proceedings of the 3rd Machine Learning for Healthcare Conference*, 383–402. PMLR.

Sanh, V.; Debut, L.; Chaumond, J.; and Wolf, T. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. *CoRR* abs/1910.01108.

Wang, T.; Zhang, Y.; Zhang, Y.; Lu, H.; Yu, B.; Peng, S.; Ma, Y.; Li, D.; et al. 2023. A hybrid model based on deep convolutional network for medical named entity recognition. *Journal of Electrical and Computer Engineering* 2023.

Yadav, V., and Bethard, S. 2018. A survey on recent advances in named entity recognition from deep learning models. In *Proceedings of the 27th International Conference on Computational Linguistics*, 2145–2158. Association for Computational Linguistics.

Yang, Z.; Dai, Z.; Yang, Y.; Carbonell, J.; Salakhutdinov, R. R.; and Le, Q. V. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In Wallach, H.; Larochelle, H.; Beygelzimer, A.; d'Alché-Buc, F.; Fox, E.; and Garnett, R., eds., *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc.

Zafarian, A.; Rokni, A.; Khadivi, S.; and Ghiasifard, S. 2015. Semi-supervised learning for named entity recognition using weakly labeled training data. In *2015 the international symposium on artificial intelligence and signal processing (AISP)*, 129–135. IEEE.

Zhang, H.; Wang, X.; Liu, J.; Zhang, L.; and Ji, L. 2023. Chinese named entity recognition method for the finance domain based on enhanced features and pretrained language models. *Information Sciences* 625:385–400.

Appendix

Models	Model 1			Model 2			Model 3			Model 4 (Ours)		
Label	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Location	.000	.000	.000	.000	.000	.000	.375	.375	.375	.925	.933	.929
User_Name	.784	.397	.527	.910	.836	.871	.859	.838	.848	.839	.729	.780
TV_Manufacturer	.667	.250	.364	.800	.500	.615	.385	.500	.435	.688	.733	.710
Service_Offering	.500	.216	.302	.800	.216	.302	.667	.286	.400	.846	.440	.579
Video_Feature	.667	.333	.444	.600	.500	.545	1.00	.455	.625	1.00	.154	.267
Networking_Manufacturer	.722	.703	.712	.676	.622	.648	.667	.839	.743	.857	.600	.706
Model_Name	.548	.283	.374	.576	.317	.409	.474	.614	.535	.500	.300	.375
Software_Manufacturer	.957	.880	.917	.957	.880	.917	.947	.900	.923	.500	.200	.286
TV_Box_Device	.773	.680	.723	.867	.520	.650	.556	.278	.370	.714	.556	.625
Software_Device	1.00	.500	.667	1.00	.333	.500	.500	.250	.333	.000	.000	.000
Networking_Feature	.500	.333	.400	.500	.167	.250	.000	.000	.000	.500	.400	.444
Version	667	.667	.667	.667	.667	.667	.600	.545	.571	1.00	.250	.400
Networking_Protocol	.750	.600	.667	.750	.600	.667	.692	.643	.667	.000	.000	.000
Network_Service_Provider	.946	.963	.955	.944	.936	.940	.885	.920	.902	.000	.000	.000
TV_Box_Manufacturer	.000	.000	.000	.400	.222	.286	.800	.308	.444	.000	.000	.000
Average	.730	.568	.622	.794	.661	.712	.743	.704	.723	.824	.727	.773

Table 6: Our models results on specific domain dataset using, the statistical model with only basic features (model 1), the statistical model with basic + orthographic features (model 2), the neural model without knowledge transfer (model 3), the neural model with knowledge transfer (model 4). Underlined scores are best scores obtained on entities taken separately and in average.

Models	BERT-Base			XLNet			RoBERTa			DistilBERT		
Label	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Location	.400	.500	.444	.556	.625	.588	.444	.500	.471	.500	.375	.429
User_Name	.830	.913	.870	.831	.863	.847	.859	.913	.885	.809	.900	.852
TV_Manufacturer	.556	.500	.526	.222	.200	.211	.500	.400	.444	.333	.300	.316
Service_Offering	.633	.679	.655	.531	.607	.567	.655	.679	.667	.618	.750	.677
Video_Feature	.556	.455	.500	.714	.455	.556	.833	.455	.588	.625	.455	.526
Networking_Manufacturer	.643	.871	.740	.600	.871	.711	.560	.903	.691	.565	.839	.675
Model_Name	.431	.636	.514	.415	.614	.495	.509	.636	.566	.528	.636	.577
Software_Manufacturer	.905	.950	.927	.950	.950	.950	.905	.950	.927	.950	.950	.950
TV_Box_Device	.563	.500	.529	.500	.500	.500	.579	.611	.595	.571	.444	.500
Software_Device	.000	.000	.000	.143	.250	.182	.000	.000	.000	.000	.000	.000
Networking_Feature	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000	.000
Version	.667	.546	.600	.615	.727	.667	.467	.636	.539	.417	.455	.435
Networking_Protocol	.688	.786	.733	.667	.857	.750	.611	.786	.688	.647	.786	.710
Network_Service_Provider	.874	.970	.919	.882	.970	.924	.890	.970	.928	.889	.960	.923
TV_Box_Manufacturer	1.00	.385	.556	.714	.385	.500	1.00	.077	.143	1.00	.231	.375
Average	.708	.782	.743	.682	.769	.723	.722	.779	.750	.714	.761	.737

Table 7: Transformer-based models results. Underlined scores are best scores obtained on entities taken separately and in average.