Named Entity Recognition for Icelandic: BERT

Benedikt Geir Jóhannesson
Reykjavík University, Menntavegur 1, 101 Reykjavík, Iceland
benediktj20@ru.is

Abstract

Introduction

Named Entity Recognition is the task of locating and classifying named entities, in an unstructured text, into predefined categories. Named entity corpuses have been developed for many languages, and recently the first named entity corpus for Icelandic, the MIM-GOLD-NER corpus, was developed and published by the Language and Voice Lab at Reykjavík University.

Objectives

Using BERT-type models for the task of named entity recognition has resulted in promising results for many languages. In this paper, the application of a multilingual BERT model on the MIM-GOLD-NER corpus was studied, with the aim of obtaining higher scores than have previously been reported.

Methods

A multilingual BERT model was trained and tested on the MIM-GOLD-NER corpus. The corpus was prepared for training and testing using three different methods. Firstly, a simple 70% and 30% split of the corpus. Secondly, a k-fold cross validation was used. Finally, a predefined split of the corpus was used, which allowed for comparison of previously highest scoring individual models.

Results

The BERT model obtained an F_1 score of 85.68, outperforming the best reported individual model by 1.78 percentage points. When compared to the best performing models for Icelandic, BERT outperformed all individual models and showed improvement in scores for six out of eight named entities types.

Conclusion

The results presented in this paper indicate that higher scores can be obtained for the task of named entity recognition by using BERT-type models. A future development and training of a BERT model specifically tailored for Icelandic will most likely yield even higher scores.

1 Introduction

Named entity recognition (NER) is a natural language processing (NLP) technique tasked with locating and classifying named entities in an unstructured text into predefined categories. NER is a very important technique for extracting relevant information from texts and has various applications. These applications include information extraction (Pokharel, 2018), question-answer systems (Toral et al., 2005), automatic summarising and machine translation (Babych and Hartley, 2003).

To give an example of NER, consider the following sentence: *Reykjavik University's research most influential according to Times Higher Education (THE) World University Ranking*. The task of NER in this example is to locate and classify both *Reykjavik University* and *Times Higher Education* as organisations.

NER systems are either a set of hand crafted rules or a machine learning system, and sometimes a combination of both. Systems relying on handcrafted rules can often by very successful. This

is especially true when operating in well defined domains (Chiticariu et al., 2010). However, constructing these rules is a very time consuming and expensive task. Also, the rules have to be manually changed each time the data changes.
With the increasing amount of data available in the world today, machine learning methods and language models have become increasingly popular to solve the task of NER.

For most NLP tasks, such as NER, a corpus is needed to do any sort of hypothesis testing, validation, and analysis. A corpus is a collection of linguistic data having the main purpose of verifying a hypothesis of a language task. NER corpuses have been developed for many languages and recently the annotation of the MIM-GOLD-NER[1] corpus was completed and published by S. Ingólfsdóttir et al. (2020). The MIM-GOLD-NER corpus contains over 1 million tokens with $48,371$ annotated named entities (NEs) of 8 different entity types: PERSON, LOCATION, ORGANISATION, MISCELLANEOUS, DATE, TIME, MONEY and PERCENTAGE.

The MIM-GOLD-NER corpus is annotated using the IOB2 format. IOB (inside-outside-beginning) tagging was presented in Ramshaw and Marcus (2002) and it is a format commonly used for named entity tagging. The IOB2 format and the IOB format are the same, except that the IOB2 format uses *B* prefixes at the beginning of every chunk.
The *I* prefix is used to indicate a tag inside a chunk, the *O* prefix is used to indicate a tag outside a chunk, and the *B* prefix is used to indicate a tag at the beginning of a chunk. An example of the IOB2 format applied to a sentence can be seen in Table 1.

As well as annotating and publishing the MIM-GOLD-NER corpus, S. Ingólfsdóttir et al. (2020) trained and evaluated four models on the corpus. Two types of bidirectional long short-term memory (BiLSTM) model, a conditional random field (CRF) model, and a perceptron model. These models were combined, into what they call the *CombiTagger*, using simple voting where the majority of votes decides on the tagging. Their paper reports an F_1-score of 85.79, making it the best performing method when published. The authors conclude the paper by expressing their optimism of obtaining higher scores by using more advanced models, such as bidirectional encoder representations from transformers (BERT).

BERT models have been used for the task of NER for many languages, Arkhipov et al. (2019), Malmsten et al. and Baumann (2019), with very promising results. BERT, introduced in 2018 by Devlin et al. (2018), applies bidirectional training of an attention model to language modelling. Language models trained this way can have a deeper sense of flow and context in texts. A pre-trained BERT model is trained to understand languages, which can then fine-tune this pre-trained model to learn specific tasks, such as NER (Wang et al., 2020).

As of today BERT presents state-of-the-art results[2] in various natural language processing tasks, including NER.

The aim of this study is to investigate the potential of using a pre-trained multilingual BERT model for the task of NER for Icelandic, as well as to see if a higher F_1 score can be obtained by applying a BERT model on the MIM-GOLD-NER corpus.

In this paper, both the pre-trained multilingual BERT model and the fine-tuning of it is described. The fine-tuned BERT models was applied on the MIM-GOLD-NER corpus. Three different evaluation approaches are described. A random train test split, a k-fold cross-validation, and finally a predefined train test split as used by S. Ingólfsdóttir et al. (2020) was applied for comparison. The results obtained are compared to the ones reported by S. Ingólfsdóttir et al. (2020) and optimism of obtaining higher F_1-scores for NER for Icelandic can indeed be confirmed.

The paper is organised as follows. Section 2 describes the methods used as well as the experiments performed. Experimental results are presented in Section 3, and the paper is concluded in Section 4.

2 Methods

2.1 BERT

Google Research[3] implements and releases pre-trained BERT models, BERT-Base and BERT-Large, either uncased or cased.

The uncased models apply lowercase to the text

[1] http://malfong.is/?pg=mim_gold_ner

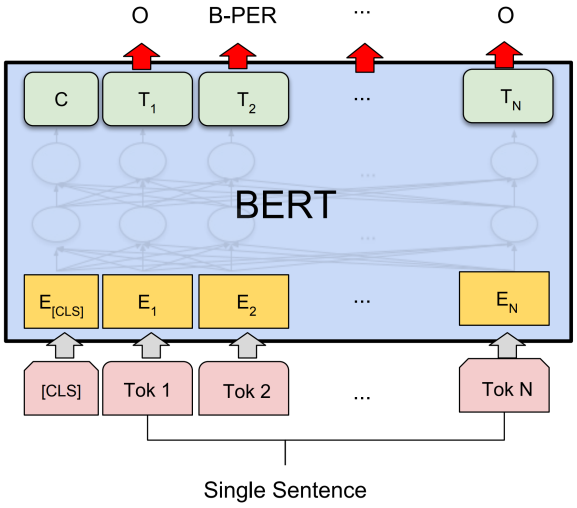
[2] https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

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

Table 1: An example of a sentence tagged with the IOB2 format.

Alex	is	going	to	Los	Angeles
B-PER	O	O	O	B-LOC	I-LOC

Figure 1: Named entity recognition modification for BERT. The input is a single sequence of tokens, *Tok N*, which are represented as embeddings *E*. *T* represents the contextual representation of a token.



at hand and strips away accent markers, while the cased models preserve both the case and the accent markers of the text. In most cases the uncased models perform better. This is mainly due to the fact that for most linguistic tasks the case information is unimportant. The opposite is the case for the task of NER where the case plays an important role.

Figure 1 demonstrates the NER modification to BERT. The input is a single sequence of tokens, *Tok N*, which are represented as embeddings, *E*. These embeddings are then fed into a neural network. *T* represents the contextual representation of a token which is finally mapped to a NE tag.

The BERT model used in this experiment was a pre-trained multilingual cased model covering 104 languages, *BERT-Base Multilingual Cased* (Pires et al., 2019).

The multilingual BERT model was pre-trained on the top 100 languages with the largest Wikipedias', including Icelandic. For training, the entire Wikipedia for each language was used, excluding the user and talk pages. As one can imagine, the sizes of these Wikipedias vary greatly. In order to prevent languages from being under-represented due to their small size, exponentially smoothed weighting of the training data was performed. This makes languages such as Icelandic over sampled when the models are trained.

The *110k shared WordPiece* (Wu et al., 2016) vocabulary was used for tokenization during the pre-training. For most languages, the following recipe is applied: lower casing and accent removal, punctuation splitting, and then whitespace tokenization.

2.2 Configuration and Environment

Configurations and environment setup for the experiments performed are listed in Table 2.

Table 2: Configurations and environment setup for experiments.

Language	Python 3.7
BERT Model	BERT-base, Multilingual Cased
Data	MIM-GOLD-NER
CPU	Intel i7-9700K, 3.6 GHz, 8 cores
GPU	GeForce RTX 2070 SUPER

The pre-trained BERT model used has 12-layers, 768-hidden layers, 12-heads and 110M parameters.

2.3 Data

As mentioned above, the MIM-GOLD-NER corpus was used for this experiment. To be able to apply the BERT model on this corpus, some minor data preparation had to be done.

The MIM-GOLD-NER corpus contains data from adjudications, blogs, books, emails, news, laws, essays, websites and texts written to be spoken. This data was separated by origin into 13 files. The initial idea was to use these files individually to train 13 domain specific models. However, the size of these individual files were not large enough for the model to be trained on. Therefore a single model was trained on the whole corpus. The first step in the data preparation was to carefully merge these files into a single file. Next, words belonging to the same sentence were grouped together and each sentence was given an

identity. This allowed the model to be trained on sentences rather than words.

The data was then tokenized using the BertTokenizer, with the added tokens '[CLS]' (classification) at the front of a sentence and '[SEP]' (separation) at the end of a sentence. An example of this tokenization can be seen in Figure 2. These tokens were added since BERT was pre-trained using the following format: '[CLS] *sentence 1* [SEP] *sentence 2* [SEP]'.

Training BERT models is a computationally expensive task and therefore a maximum length of 75 characters per sentence was set. All sentences, and their appropriate tag sequences, were either trimmed or padded to meet the condition of the maximum length.

Table 3 and 4 contain some basic information on the prepared data, as well as the distribution of NE types in the MIM-GOLD-NER corpus.

Table 3: Token and sentence count in the MIM-GOLD-NER corpus.

Token count:	1,005,688
Unique token count:	106,524
Unique tag count:	17
Sentence count:	49,527

Table 4: Individual names entities count and statistics in the MIM-GOLD-NER corpus.

Type	**Count**	**Ratio (%)**
Person	15,599	32.3
Location	9,011	18.6
Organization	8,966	18.5
Miscellaneous	6,264	13.0
Date	5,529	11.4
Time	1,214	2.5
Money	1,050	2.2
Percent	738	1.5
Total	**48,371**	**100**

2.4 Experiment

Once the data had been prepared, the actual experimentation was conducted.

PyTorch, a machine learning library, was used for the implementation, therefore the data needed to be transformed into a special format called torch tensors. Once transformed, the sentences were shuffled and dataloaders defined to be trained on.

Next, the model was fine-tuned using the AdamW optimizer (Loshchilov and Hutter, 2019). The model was then fitted to the training data using five epochs and train loss was collected per epoch.

The running time of the training depends heavily on the hardware used, especially the GPU (Raina et al., 2009), and the number of epochs. The training of the model ranged from 20 to 40 minutes, depending on the evaluation method used, using the setup described above. When the training was completed, the trained model was saved for potential future usage, and evaluated.

The evaluation metrics introduced at CoNLL-2003 (Tjong Kim Sang and De Meulder, 2003) were used to evaluate and measure performance of the model. Precision is the percentage of NEs correctly identified by the model. Recall is the percentage of NEs present in the corpus that are identified by the model. F_1 is the combination of precision and recall as a harmonic mean. Support is the number of occurrences of each NE class in the test data.

For the first experiment, a simple random splitting of the data was performed, 70% of the data was used for training and 30% for testing.

For the second experiment, a k-fold cross-validation method was used to estimate the performance of models.
The sentences were shuffled randomly and split up into 10 groups, $k = 10$. For each of these groups, the group in question was reserved as test data and the model was trained using the rest of the data. The performance of the model was then evaluated for each group. The model was discarded after each evaluation.
After performing this for each of the 10 groups, the average of all the F_1-scores collected was calculated (Tjong Kim Sang and De Meulder, 2003). Using a k-fold cross-validation is considered a good method of estimating performance of models as it generally has lower bias than other evaluation methods (Kohavi, 2001).

For the third, and last experiment, a predefined split of the data was used. This is the same split as used by S. Ingólfsdóttir et al. (2020). By using this predefined split, a comparison between results presented in this paper and results presented by S. Ingólfsdóttir et al. (2020) could be made.
The training data contained $805,748$ tokens and

Figure 2: Example of tokenization using the BertTokenizer.

[CLS] Dr . Hannes H ##ög ##ni Vi ##l ##h ##já ##lm ##sson hefur hl ##oti ##ð fram ##gang ...

39,467 sentences while the test data contained 100,693 tokens and 4,959 sentences. It was important that the ratio of sentences was similar to the ratio of tokens between training and testing for consistency.
For this experiment some minor changes were made to the implementation. This modified code is publicly available on GitHub[4].

3 Results

In this section the results obtained from the three different training setups on the MIM-GOLD-NER corpus, explained earlier, are presented.

For the first experiment, the data was split into training data and testing data, containing 70% and 30% of the data respectively. The training data contained 705,030 tokens, and the testing data contained 302,100 tokens.

An F_1-score of 88.37 and an accuracy score of 99.02 was obtained using this setup.
The model was trained and tested 10 times, using a new split each time, and the results presented are the average of those results. Table 5 shows detailed results for individual NE types from training and testing the model using the traditional split described.

For each evaluation phase in the second experiment, the training data contained 906,441 tokens and the testing data contained 100,693 tokens.

For each iteration both the F_1-score and the accuracy score were collected, and the averages computed. An average F_1-score of 89.24 and an average accuracy score of 99.06 were obtained. Table 6 shows the average F_1-scores for individual NE types.

Individual evaluation reports are publicly available on GitHub[5].

Finally, for the third and last experiment, a predefined split of the corpus, as explained above, was used. The training data contained 805,748 tokens and the testing data contained 100,693 tokens.

Using this setup an F_1-score of 85.68 and an accuracy score of 98.52 was obtained. Table 7 shows detailed results for individual NE types from training and testing the model using the predefined split.

Table 8 shows a detailed comparison between results obtained in this paper and results presented in (S. Ingólfsdóttir et al., 2020).

3.1 NER API

As a result of this experimentation, an application programming interface (API) for the trained model was introduced[6].

This API runs a pre-trained BERT model, fine tuned for the task of NER for Icelandic as described in this paper. It should be noted that this API is an experimental work in progress and might therefore be unstable at times.

The model described in section 2.4 was saved and made available through this API.
There are some limits to this API. It only annotates the first 75 characters of a sentence, and each API call has an execution time of around 20 to 30 seconds since the model operates on a CPU instead of a GPU.

The API takes in a *query*, a sentence from a user. The query is then tokenized, preprocessed and passed to the model. The model annotates the query and returns a response containing each word of the query paired with the corresponding NE type.

The API is hosted on pythonanywhere[7] and all the code is publicly available on GitHub[8].

4 Discussion

In this paper, the experimentation and application of a fine-tuned *BERT-Base Multilingual Cased* model on the recently published MIM-GOLD-NER corpus was described. The BERT model was fine-tuned for the task of NER for Icelandic. Different evaluations were presented and a comparison to the current best performing models for the task of NER for Icelandic was made. Most significant results are the ones obtained when a predefined split of the corpus was used, as well as

[4] https://github.com/bennigeir/NER
[5] https://github.com/bennigeir/NER/tree/main/code/data/results/10-cross
[6] http://www.ice-bert-ner.com
[7] http://www.pythonanywhere.com
[8] https://github.com/bennigeir/NER

Table 5: Precision, recall, F_1-score, and support per named entity type from 70% training and 30% testing split for the multilingual BERT model.

	Precision	**Recall**	**F_1-score**	**Support**
Person	92.67	95.10	93.87	4,371
Location	90.41	92.61	91.50	2,504
Organisation	84.55	81.43	82.97	2,467
Miscellaneous	73.86	78.67	76.19	1,810
Date	89.36	91.23	90.28	1,528
Time	88.83	91.20	90.00	375
Money	84.30	86.36	85.32	286
Percent	84.42	97.13	95.75	209

Table 6: Average of F_1-scores per named entity type using 10-fold cross validation for the multilingual BERT model.

	Average F_1-score
Person	92.90
Location	92.28
Organisation	83.26
Miscellaneous	77.10
Date	94.46
Time	89.63
Money	89.04
Percent	95.85

when a *K*-fold cross-validation was used. The results reported in this paper are currently the highest obtained by a single model for the task of NER for Icelandic. The BERT model outperforms the BiLSTM-GloVE model, which is the most advanced and highest scoring model used by S. Ingólfsdóttir et al. (2020), by 1.78 percentage points.

When the results obtained were compared to results presented in S. Ingólfsdóttir et al. (2020), it was evident that BERT outperformed individual models overall. BERT outperformed all other individual models in six out of eight types of NEs. Moreover, BERT outperformed the CombiTagger, a combination of three highest scoring individual models, in four out of eight types of NEs. This outperformance of BERT, when compared to individual models, was expected since BERT presents state-of-the-art results[9] in various natural language processing tasks.

[9] https://ai.googleblog.com/2018/11/open-sourcing-bert-state-of-art-pre.html

The same can be said for the overall outperformance by the CombiTagger. It has been found that in general an ensemble of good performing models outperform a single model (Maclin and Opitz, 2011).

Both interesting and surprising observations were made when the results are examined further. Firstly, is that the BERT model outperformed the CRF model for the LOCATION and ORGANISATION entity types. This is surprising since the CRF model is implemented using a lookup-based method, which searches a pre-stored list of locations and organisations, and tends to increase performance. Secondly, is the performance of BERT for the MISCELLANEOUS entity type. An F_1-score of 71.17 is obtained, which is 9.40 percentage points higher than the highest scoring individual model.

The limitations of this study include the fact that no experiments were done which combined BERT and other high performing models. High performing models are often combined using an ensemble method in order to obtain better results.
Also, another popular method used to obtain better results is the post processing method of searching a pre-stored list of locations and organisations. The results in this paper are reported without using any post processing methods.

The main strength of this study is that this is the first attempt to explore and apply a BERT-type model for the task of NER for Icelandic, and the results indicate good performance.

In conclusion, a fine-tuned *BERT-Base Multilingual Cased* model on the recently published MIM-GOLD-NER corpus. Three different training and testing setups of the corpus were used to

Table 7: Precision, recall, F_1-score, and support per named entity type from the predefined train-test split for the multilingual BERT model.

	Precision	**Recall**	**F_1-score**	**Support**
Person	90.51	91.31	90.91	1,243
Location	84.39	87.91	86.11	984
Organisation	80.90	80.59	80.74	1,293
Miscellaneous	69.38	73.04	71.17	664
Date	91.42	91.94	91.68	707
Time	95.95	95.13	95.54	349
Money	86.61	91.51	88.99	106
Percent	99.06	100.00	99.53	105

Table 8: Comparison of F_1-scores per named entity type between results obtained by applying multilingual BERT model on the MIM-GOLD-NER corpus, and results presented in S. Ingólfsdóttir et al. (2020).

	BERT-Base	**CombiTagger**	**BiLSTM-GloVE**	**BiLSTM-internal**	**CRF**	**IXA**
Person	**90.91**	90.19	89.53	80.11	87.18	87.90
Location	86.11	**88.21**	85.45	74.98	86.04	85.54
Organisation	80.74	**81.23**	79.03	65.85	77.02	77.02
Miscellaneous	**71.17**	64.27	61.77	44.10	53.87	61.57
Date	91.68	**93.13**	90.60	86.73	90.90	91.96
Time	95.54	**96.41**	94.78	91.83	94.46	94.29
Money	88.99	86.58	**89.45**	81.86	84.30	85.59
Percent	**99.53**	98.65	95.54	92.73	98.21	97.35
Overall	85.68	**85.79**	83.90	73.60	82.24	83.10

evaluate the model. The results in this paper indicate that higher scores can be obtained by using BERT models.

Further research and experimentation, using BERT-type models, combining results of multiple models, and the exploitation of lookup-based method is believed to further increase the likelihood of obtaining higher scores in the future.

The greatest improvement will most likely be achieved by training a BERT model specifically for Icelandic.

References

Mikhail Arkhipov, Maria Trofimova, Yuri Kuratov, and Alexey Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition. pages 89–93.

Bogdan Babych and Tony Hartley. 2003. Improving machine translation quality with automatic named entity recognition.

Antonia Baumann. 2019. Multilingual language models for named entity recognition in german and english. pages 21–27.

Laura Chiticariu, Rajasekar Krishnamurthy, Yunyao Li, Frederick Reiss, and Shivakumar Vaithyanathan. 2010. Domain adaptation of rule-based annotators for named-entity recognition tasks. pages 1002–1012.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding.

Ron Kohavi. 2001. A study of cross-validation and bootstrap for accuracy estimation and model selection. 14.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled weight decay regularization.

Richard Maclin and David W. Opitz. 2011. Popular ensemble methods: An empirical study. *CoRR*, abs/1106.0257.

Martin Malmsten, Love Börjeson, and Chris Haffenden. Playing with words at the national library of sweden : Making a swedish BERT.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? pages 4996–5001.

Prabhat Pokharel. 2018. *Information Extraction Using Named Entity Recognition from Log Messages*. Ph.D. thesis.

Rajat Raina, Anand Madhavan, and Andrew Y. Ng. 2009. Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, page 873–880, New York, NY, USA. Association for Computing Machinery.

Lance Ramshaw and Mitchell Marcus. 2002. Text chunking using transformation-based learning. *Third ACL Workshop on Very Large Corpora. MIT*.

S. Ingólfsdóttir et al. 2020. Named entity recognition for icelandic: Annotated corpus and models.

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.

Antonio Toral, Elisa Noguera, Fernando Llopis, and Rafael Muñoz. 2005. Improving question answering using named entity recognition. 3513:181–191.

Yu Wang, Yining Sun, Zuchang Ma, Lisheng Gao, Yang Xu, and Ting Sun. 2020. Application of pre-training models in named entity recognition.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.