# Named Entity Recognition for Icelandic: BERT

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

## Abstract

Named Entity Recognition (NER) is the task of locating and classifying named entities (NEs) in an unstructured text into predefined categories.

NER datasets have been developed for many languages and recently the first NER corpus for Icelandic, MIM-GOLD-NER, was created and published by Reykjavík University. The MIM-GOLD-NER corpus contains over 1 million tokens with $48,371$ annotated named entities.

In this research report we study the application of multilingual BERT model on the MIM-GOLD-NER corpus. This model obtains an $F_1$ score of $84.95$. When compared with the best performing models for Icelandic, BERT outperforms individual models.

As a result from this report a NER API was created and made publicly available.

The future for NER for Icelandic is bright with BERT.

## 1 Introduction

Named Entity Recognition (NER) is the task of locating and classifying named entities in an unstructured text into predefined categories. NER is an important technique for extracting relevant information from texts and it has various applications. These applications include information extraction, question-answer systems (Toral et al., 2005), automatic summarizing and machine translation (Babych and Hartley, 2003).

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

NER datasets have been developed for many languages and recently the annotation of the MIM-GOLD-NER 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, organization, 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 chunks. 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.

BERT models have been used for the task of NER for many languages, (Arkhipov et al., 2019), (Malmsten et al., 2020) and (Baumann, 2019), with very promising results.

In this research report we describe the pretrained multilingual BERT model and the fine-tuning of it to handle the task of NER for Icelandic. We then apply this fine-tuned model on the MIM-GOLD-NER and we obtain an $F_1$ score of $84.95$. We also describe three different evaluation approaches. We then compare the results obtained with the ones reported by (S. Ingólfsdóttir et al., 2020) and we confirm that we can indeed be optimistic on obtaining higher $F_1$-scores for NER for Icelandic than previously reported.

Finally, as a results of this report, we introduce a NER API which runs our pre-trained BERT model fine tuned for Icelandic. This API is publicly available and can be accessed here.

The code for both the API and the models is publicly available on GitHub[1].

## 2 Related Work

NER systems are either a set of hand crafted rules or a machine learning system, and sometimes a combination of both. Systems relying on hand-crafted rules can often by very successful. This is especially true when dealing with well defined domains (Chiticariu et al., 2010). However, constructing these rules is a time consuming task, it is expensive and the rules have to be changed manually each time the data changes.
With the increasing amount of data available in the world today, machine learning methods have become more and more popular and more available.

As well as annotating and publishing the MIM-GOLD-NER corpus, (S. Ingólfsdóttir et al., 2020) trained and evaluated three models on the corpus. A BiLSTM model, a CRF model, and a perceptron model. These models were then combined, into what they call the *CombiTagger*, using simple voting. The paper reports an $F_1$-score of 85.79 and the authors express their optimism of obtaining higher scores by using more advanced models, such as BERT.

BERT was introduced in 2018 by (Devlin et al., 2018). BERT stands for bidirectional encoder representations from transformers.
BERT 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. We can then fine-tune this pre-trained model to learn specific tasks, such as NER.

As of today BERT presents state-of-the-art results in various natural language processing tasks, including NER. Figure 1 demonstrates the NER modification for BERT.

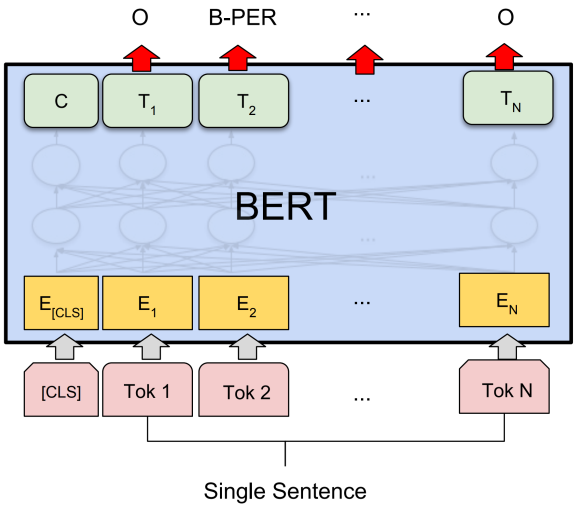


Figure 1: NER modification for BERT.

## 3 Experiment Details

### 3.1 BERT

Google Research[2] releases pre-trained BERT models, BERT-Base and BERT-Large, either uncased or cased.
The uncased models apply lowercase the text and strips away accent markers while the cased models preserve the case and accent markers. Most often the uncased models perform better, except when the case information is important. This is the case for the task of NER.
The BERT model we use is a pre-trained multilingual BERT model covering 104 languages, *BERT-Base, Multilingual Cased* (Pires et al., 2019).
The multilingual BERT models are pre-trained on the top 100 languages with the largest Wikipedias. Fortunately Icelandic is one of these languages. For training, the entire Wikipedia for each language was used, excluding user and talk pages. As we can imagine the size of these Wikipedias vary greatly in size. In order to prevent languages from being underrepresented, exponentially smoothed weighting of the training data was performed. This makes languages such as Icelandic over sampled.
The *110k shared WordPiece* (Wu et al., 2016) vocabulary is 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.

[1] https://github.com/bennigeir/NER
[2] https://github.com/google-research/bert/

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

Table 1: Example of IOB2 format.

### 3.2 Configuration and Environment

For our experiment the following configurations and environment were used:

| | |
|---|---|
| Language | Python 3.7 |
| GPU | GeForce RTX 2070 SUPER |
| BERT Model | BERT-base, Multilingual Cased |
| Dataset | MIM-GOLD-NER |

The pre-trained BERT model has 24-layers, 1024-hidden, 16-heads and 340M parameters.

### 3.3 Data

As mentioned before we use the MIM-GOLD-NER corpus for this experiment. To be able to use the BERT model on this corpus some minor data preparation had to be done.

The MIM-GOLD-NER corpus contains data from adjudication, blog, books, emails, news, laws, essays, websites and text written to be spoken. This data is separated by origin into 13 files. Our first step was to carefully merge these files into one. The idea was to use individual files to train a domain specific model. However the sizes of individual files were not large enough for this to be done.

Next we group words belonging to the same sentence together and we give each sentence an id.

The dataset is then tokenized using the BertTokenizer with the added tokens '[CLS]' at the front of a sentence and '[SEP]' at the end of a sentence. An example of this tokenization can be seen in Table 2. This is done since BERT is pre-trained using the following format: '[CLS] *sentence 1* [SEP] *sentence 2* [SEP]' ...

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

Following are some basic information on the prepared dataset as well as the distribution of NE types in the MIM-GOLD-NER corpus:

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

| **Type** | **Count** | **Ratio** |
|---|---|---|
| Person | 15,599 | 32.25% |
| Location | 9,011 | 18.63% |
| Organization | 8,966 | 18.54% |
| Miscellaneous | 6,264 | 12.95% |
| Date | 5,529 | 11.43% |
| Time | 1,214 | 2.51% |
| Money | 1,050 | 2.17% |
| Percent | 738 | 1.53% |
| **Total** | **48,371** | **100%** |

### 3.4 Experiment

Once the dataset has been prepared we proceed with the actual experimentation.

Since we are using PyTorch for our implementation we transform the dataset into torch tensors. The dataset is then shuffled and we define the dataloaders.

Next we fine-tune the model using the AdamW optimizer (Loshchilov and Hutter, 2019). We then fit the model to our training dataset in 5 epochs and collect train loss per epoch.

The running time of the training depends heavily on the hardware used, especially the GPU. It took between 20 and 40 minutes to train the model on the training data using the GPU mentioned above.

When the training is completed we save the trained model, for potential future usage, and proceed to evaluate it.

For our first experimentation we perform a simple splitting of the dataset, 70% for training and 30% for testing. From this experiment we obtain an $F_1$-score of 88.37. See evaluation details in section 4.

For the second experiment we use 10-fold cross-validation. Cross-validation is a popular method to estimate the performance of models.

We start by shuffling the sentences randomly and

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

Table 2: Example of BERT tokenization

split them up into 10 groups. For each group we then reserve that group as a test dataset and train using the rest of the dataset. We then proceed to evaluate the performance of the model and make sure to discard the model when the evaluation is finished.
After performing this for each of the 10 groups, we calculate the average of all the $F_1$-scores collected.
From this experiment we obtain an $F_1$-score of $89.45$. See evaluation details in section 4.

For the last experiment we use a predefined split of the dataset. This is the same split as used by (S. Ingólfsdóttir et al., 2020). By doing this we are able to compare our results with the results obtained and presented in that paper. This serves as a good benchmark for our model.
The training dataset contains $805,748$ tokens and $39,467$ sentences while the test dataset contains $100,693$ tokens and $4,959$ sentences. It is important that we make sure that the ratio of sentences is similar to the ratio of tokens between training and testing.
For this experiment we make some minor changes to the implementation since the dataset has been split. This modified code is available on GitHub[3].
From this experiment we obtain an $F_1$-score of $85.68$. See evaluation details in section 4.

## 4 Evaluation

In this section we present and discuss the results gathered from three different training setups on the MIM-GOLD-NER corpus.

For our first experiment we split the corpus into training and testing, 70% and 30% respectively. Table 3 shows detailed results for individual NE types from training and testing BERT using the split described. We obtain an $F_1$-score of $88.37$ and an accuracy score of $99.02$.
The model was trained multiple times on different splits and the results were all consistent with each other.

For our second experiment we applied a 10-fold cross-validation to estimate performance. For each iteration we gathered both the $F_1$-score and the accuracy score. The average for both of them were then computed. We obtain an average $F_1$-score of $89.24$ and an average accuracy score of $99.06$. K-fold cross-validation results generally have lower bias than other evaluation methods and are therefore considered a good method of estimating performance.
Individual evaluation reports can be found on GitHub[4].
Finally, for our last experiment we use a predefined split of the dataset as explained above. Table 4 shows detailed results for individual NE types from training and testing BERT using the predefined split. We obtain an $F_1$-score of $85.68$ and an accuracy score of $98.52$.

If we compare these results to the ones presented in (S. Ingólfsdóttir et al., 2020) it seems that BERT outperforms individual models overall. BERT outperforms the other models in 6 out of 8 individual types of named entities. It even outperforms the CombiTagger in 4 out of 8 types.
Most surprising is the performance of BERT for the miscellaneous entities, an $F_1$-score of $71.17$, $6.9$ percentage points higher than the highest scoring individual model. A detailed comparison between results obtained using BERT and results from models presented in (S. Ingólfsdóttir et al., 2020) are shown in 5.

## 5 NER API

As a result of this project we introduce an application programming interface[5] (API) for BERT. This API runs a pre-trained BERT model fine tuned for the task of NER for Icelandic. It should be noted that this API is a work in progress and might be unstable at times.

The model described in Section 3.4 has been 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 we operate on a CPU instead of a GPU.

[3] https://github.com/bennigeir/NER

[4] https://github.com/bennigeir/NER/tree/main/code/data/results/10-cross

[5] http://www.ice-bert-ner.com

| NE type | Precision | Recall | $F_1$-score | Support |
|---|---|---|---|---|
| **Date** | 0.8936 | 0.9123 | 0.9028 | 1528 |
| **Location** | 0.9041 | 0.9261 | 0.9150 | 2504 |
| **Miscellaneous** | 0.7386 | 0.7867 | 0.7619 | 1810 |
| **Money** | 0.8430 | 0.8636 | 0.8532 | 286 |
| **Organization** | 0.8455 | 0.8143 | 0.8297 | 2467 |
| **Percent** | 0.8442 | 0.9713 | 0.9575 | 209 |
| **Person** | 0.9267 | 0.9510 | 0.9387 | 4371 |
| **Time** | 0.8883 | 0.9120 | 0.9000 | 375 |

Table 3: Evaluation results per NE type from 70% training and 30% testing split.

| NE type | Precision | Recall | $F_1$-score | Support |
|---|---|---|---|---|
| **Date** | 0.9142 | 0.9194 | 0.9168 | 707 |
| **Location** | 0.8439 | 0.8791 | 0.8611 | 984 |
| **Miscellaneous** | 0.6938 | 0.7304 | 0.7117 | 664 |
| **Money** | 0.8661 | 0.9151 | 0.8899 | 106 |
| **Organization** | 0.8090 | 0.8059 | 0.8074 | 1293 |
| **Percent** | 0.9906 | 1.0000 | 0.9953 | 105 |
| **Person** | 0.9051 | 0.9131 | 0.9091 | 1243 |
| **Time** | 0.9595 | 0.9513 | 0.9554 | 349 |

Table 4: Evaluation results per NE type from the predefined train-test split.

| NE type | BERT-Base | CombiTagger | BiLSTM-GloVE | BiLSTM-internal | CRF | IXA |
|---|---|---|---|---|---|---|
| **DATE** | 0.9168 | 0.9313 | 0.9060 | 0.8673 | 0.9090 | 0.9196 |
| **LOC** | 0.8611 | 0.8821 | 0.8545 | 0.7498 | 0.8604 | 0.8554 |
| **MISC** | 0.7117 | 0.6427 | 0.6177 | 0.4410 | 0.5387 | 0.6157 |
| **MON** | 0.8899 | 0.8658 | 0.8945 | 0.8186 | 0.8430 | 0.8559 |
| **ORG** | 0.8074 | 0.8123 | 0.7903 | 0.6585 | 0.7702 | 0.7702 |
| **PERC** | 0.9953 | 0.9865 | 0.9554 | 0.9273 | 0.9821 | 0.9735 |
| **PER** | 0.9091 | 0.9019 | 0.8953 | 0.8011 | 0.8718 | 0.8790 |
| **TIME** | 0.9554 | 0.9641 | 0.9478 | 0.9183 | 0.9446 | 0.9429 |
| **OVERALL** | **0.8568** | **0.8579** | **0.8390** | **0.7360** | **0.8224** | **0.8310** |

Table 5: Comparison of $F_1$-scores between results obtain using BERT and results from models presented in (S. Ingólfsdóttir et al., 2020).

The API takes in a *query*, the query is then tokenized and passed to our model. The model annotates the query and returns a JSON response containing each word of the query paired with the corresponding entity type.

The API is hosted on pythonanywhere[6] and all the code is publicly available on GitHub[7].

Example response from the API:

```
"results": [
    [
        "[CLS]",
        "[CLS]"
    ],
    [
        "Erna",
        "B-Person"
    ],
    [
        "Sif",
        "I-Person"
    ],...
```

[6] http://www.pythonanywhere.com
[7] https://github.com/bennigeir/NER

## 6 Conclusion

In this research report we have described how we applied the *BERT-Base Multilingual Cased* model on the MIM-GOLD-NER corpus.

We presented different evaluations and compare our results with the best performing models for Icelandic.

Using advanced models, such as BERT, for the task of NER for Icelandic does indeed result in higher $F_1$-scores than previous individual models.

Further research and usage would be very beneficial for this task.

## 7 Acknowledgements


- Hrafn Loftsson for providing predefined split of the MIM-GOLD-NER corpus. This allowed us to compare results.

- Google for hosting BERT models and making them publicly available.


## References


Mikhail V. Arkhipov, Maria Trofimova, Yuri Kuratov, and A. Sorokin. 2019. Tuning multilingual transformers for language-specific named entity recognition.

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.

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

Martin Malmsten, Love Börjeson, and Chris Haffenden. 2020. Playing with words at the national library of sweden - making a swedish bert. *ArXiv*, abs/2007.01658.

Telmo Pires, Eva Schlinger, and Dan Garrette. 2019. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy. Association for Computational Linguistics.

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.

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

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