UNIVERSITY OF HELSINKI
FACULTY OF ARTS
DEPARTMENT OF DIGITAL HUMANITIES

Master's Thesis

Multilingual Named Entity Recognition through Data and Model Transfer

Saara Palma-Suominen
014291850

Master's Programme in Linguistic Diversity and Digital Humanities

Supervisors: Jörg Tiedemann, Krister Lindén 06.04.2021

Tiivistelmä

Tiedekunta: Humanistinen tiedekunta

Koulutusohjelma: Kielellisen diversiteetin ja digitaalisten ihmistieteiden maisteriohjelma

Opintosuunta: Kieliteknologia

Tekijä: Saara Palma-Suominen

Työn nimi: Multilingual Named Entity Recognition through Data and Model Transfer

Työn laji: Maisterintutkielma

Kuukausi ja vuosi: Huhtikuu 2021

Sivumäärä: Tutkielma 43 sivua, liitteet 12 sivua

Avainsanat: NLP, Language Technology, NER, Named Entity Recognition, multilingual, BERT, data transfer, model transfer, kieliteknologia, nimien tunnistus

Säilytyspaikka: Helsingin yliopiston kirjasto

Muita tietoja:

Tiivistelmä:

Maisterintutkielma käsittelee monikielistä nimien tunnistusta. Tutkielmassa testataan kahta lähestymistapaa monikieliseen nimien tunnistukseen: annotoidun datan siirtoa toisille kielille, sekä monikielisen mallin luomista. Lisäksi nämä kaksi lähestymistapaa yhdistetään. Tarkoitus on löytää menetelmiä, joilla nimien tunnistusta voidaan tehdä luotettavasti myös pienemmillä kielillä, joilla annotoituja nimientunnistusaineistoja ei ole suuressa määrin saatavilla.

Tutkielmassa koulutetaan ja testataan malleja neljällä kielellä: suomeksi, viroksi, hollanniksi ja espanjaksi. Ensimmäisessä metodissa annotoitu data siirretään kieleltä toiselle monikielisen paralleelikorpuksen avulla, ja näin syntynyttä dataa käytetään neuroverkkoja hyödyntävän koneoppimismallin opettamiseen. Toisessa metodissa käytetään monikielistä BERT-mallia. Mallin koulutukseen käytetään annotoituja korpuksia, jotka yhdistetään monikieliseksi opetusaineistoksi. Kolmannessa metodissa kaksi edellistä metodia yhdistetään, ja kieleltä toiselle siirrettyä dataa käytetään monikielisen BERT-mallin koulutuksessa.

Kaikkia kolmea lähestymistapaa testataan kunkin kielen annotoidulla testisetillä, ja tuloksia verrataan toisiinsa. Metodi, jossa rakennettiin monikielinen BERT-malli, saavutti selkeästi parhaimmat tulokset nimien tunnistamisessa. Neuroverkkomallit, jotka koulutettiin kielestä toiseen siirretyillä annotaatioilla, saivat selkeästi heikompia tuloksia. BERT-mallin kouluttaminen siirretyillä annotaatioilla tuotti myös heikkoja tuloksia.

Annotaatioiden siirtäminen kieleltä toiselle osoittautui haastavaksi, ja tuloksena syntynyt data sisälsi virheitä. Tulosten heikkouteen vaikutti myös opetusaineiston ja testiaineiston kuuluminen eri genreen. Monikielinen BERT-malli on tutkielman mukaan testatuista parhaiten toimiva metodi, ja sopii myös kielille, joilla annotoituja aineistoja ei ole paljon saatavilla.

Contents

1 Introduction

Multi-lingual approaches to Natural Language Processing have been one of the most researched and most useful innovations in the field in recent years. They are already extensively used in many natural language processing tasks such as machine translation, classification and text mining. The appeal of multi-lingual methods lies in their usefulness for both research purposes, and for the purposes of industry.

In research, multi-lingual methods help in moving away from the dominance of English in language technology. As researchers develop well-performing multi-lingual models, we gain access and tools for research also for low-resource languages. For many smaller languages, annotated data or other resources are not available at least in large amounts. Multi-lingual systems can help overcome this issue, and aid in developing tools for analyzing small languages as well.

My inspiration for the theses was to explore multi-lingual methods and find ways to conduct named entity recognition in low-resource settings. Among other languages, I chose to work with Finnish and Estonian, which can be considered as relatively low-resource languages compared to many other European languages. However, on a global scale these languages cannot be considered low-resource, as they have millions of speakers and an official status as both spoken and written languages. I also aimed to explore techniques which would be useful for minority languages as well, where annotated data may not exist at all.

From the industry perspective, multi-lingual systems are a crucial advantage, as many companies that specialize in text analytics and natural language processing operate in a global market. There, well-performing robust multi-lingual systems are a clear asset, as they demand much less maintenance and are easily scalable to new market areas.

In this thesis, I apply multi-lingual methods to Named Entity Recognition, which refers to extracting names, such as person names or locations, from unstructured textual data. Named Entity Recognition is an important field within NLP, and it is useful as a part of many tasks such as information extraction and text mining. It is also an interesting field to apply multi-lingual methods on, as it has properties which make it especially suitable for multi-lingual methods, but there are challenges as well. Semantically, named entities always point to a certain instance, for example to a person, building, institution or a fictional character. The same entity in several languages points to the same instance, and there is less ambiguity than in many other aspects of language. On the other hand, many named entities are inherently local and refer to people and locations which are not known in other areas, and thus may not occur in other languages

In this thesis, I test two commonly used methods of applying multi-lingual methods to Named Entity Recognition. The first of these is annotation projection, where the models built are mono-lingual, but the data is projected from one annotated data set to several other languages. The second one is building a multi-lingual transformer model, namely multi-lingual

BERT. Finally, I'll attempt to combine the two methods, and use data created by annotation projection to fine-tune the multi-lingual BERT model.

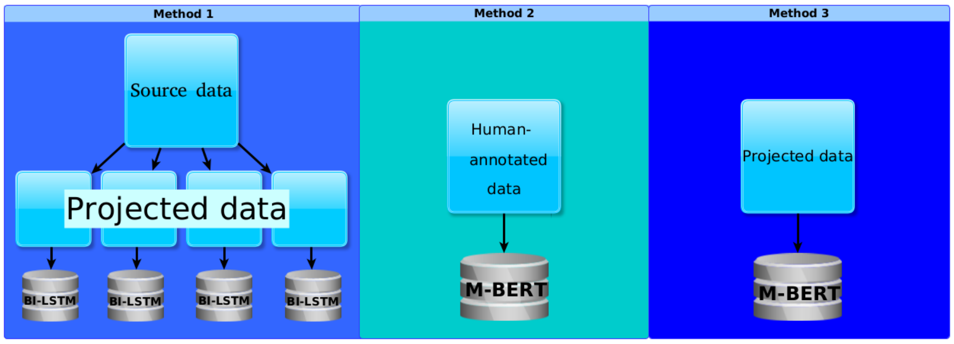


Figure 1: The methods used in the thesis

My research questions are the following:

1. How well does the annotation projection approach work compared to multi-lingual BERT?

2. Does using projected annotations to fine-tune the BERT model achieve results comparable to the same model fine-tuned with original annotated data?

2 Background

2.1 NER and multilingual methods

Named Entity Recognition and its uses

According to Nathan and DEVI (2019), the term Named Entity Recognition (NER) was first coined in the Message Understanding Conference (MUC). It was added as a task in the sixth MUC conference, which took place in 1995 (Grishman and Sundheim, 1996). The purpose of NER is to recognize names, such as person names, organizations, place names, products or events, from unstructured text. Nathan and DEVI (2019) note that NER has gained importance in the recent years, as companies and academia have recognized the potential of the huge amount of unstructured web data available (Nathan and DEVI, 2019, p.1). To efficiently use such data, one needs robust information extraction systems. NER therefore often acts as the first step of Information Extraction, but is also used in connection with other NLP tasks, such as Machine Translation, Sentiment Analysis, Question Answering and Text Summarization, among others (Nathan and DEVI, 2019, p.1).

Cross-lingual vs. multi-lingual

As opposed to mono-lingual methods, this thesis deals with multi-lingual or cross-lingual methods. These two terms are sometimes used interchangeably, but there are slight differences to their meaning. The term ‘cross-lingual’ may be used to describe systems where only two languages are used, but is also sometimes used to describe any system where language knowledge is transferred between languages. ‘Multi-lingual’ is used in a largely similar way, but often describes systems where more than two languages are used. In this thesis, both terms are used, but I will generally use ‘multi-lingual’.

Multi-lingual methods in NLP

Using multi-lingual methods in NLP has been increasingly popular in recent years. The advantages are obvious: instead of having to train several models, one for each language, one can save time and resources by building models which can be trained and used on data in several languages. It also improves scalability of systems, and can aid in solving the problem caused by a lack of annotated training data in many smaller languages. Building robust multi-lingual models for a multitude of Natural Language Processing tasks has become significantly easier in the last couple of years, with the introduction of innovations such as the multilingual version of BERT (Devlin et al., 2018). It is a language model pre-trained on Wikipedia data of the biggest Wikipedia languages. The model can then be fine-tuned with a smaller amount of annotated data to perform many kinds of language processing tasks, including classification as well as sequence labeling tasks like NER. This means one does not have to build a model from scratch, as the off-the-shelf pre-trained BERT model already contains a lot of general knowledge about languages, and can solve many kinds of NLP problems with state-of-the art results.

2.2 Previous approaches to multi-lingual NER

I will refer to the two main approaches to multi-lingual NER as *data transfer based methods* and *model transfer based methods*, following a formulation by (Wu et al., 2020, p.1). In its most general form, data transfer refers to methods where one constructs a set of named entity labels for one language, based on the existing labels in another language. (Wu et al., 2020, p.1) This can be done using a variety of methods for projecting, translating or mapping the entity labels to their counterparts in the other language.

In data transfer methods, the transfer occurs typically before modeling. However, in model transfer based methods, a model is trained on data in the source language or languages, and the model is then applied to the target language or languages (Wu et al., 2020, p.1). Perhaps the most popular approach is to use multilingual embeddings, such as multilingual BERT described in the previous chapter. Other approaches include more complex model architectures, where some or all layers or features of a neural network are shared between languages. There are also systems where several models are trained on several languages, and some kind of an algorithm is used to select the best-performing model for each example in a multi-lingual test data set.

2.3 Data transfer methods

Machine translation

One prominent example of a data transfer method has been to employ machine translation to create data in the target language or languages. For example Jain et al. (2019) used an existing English-language named entity-annotated data set, and then used Google Translate to translate the annotated text to the target languages. They then generated a list of Named Entity candidates from the target texts, and finally used matching algorithms to map the English names to the correct candidates in the target languages. This method's good sides include that no other resources are mandatory except for a robust MT system between the language pair or pairs. However, achieving the exact correct named entity in the target may be difficult due to to the inherently ambiguous nature of natural languages. A translation may be semantically correct, but still not be formally equal to the correct named entity in the target language. An example of tackling this issue is presented in Ehrmann et al. (2011), where a multilingual named entity database is used alongside a statistical machine translation system. If a source language named entity is found in the database, its counterpart in the target language can be easily and reliably retrieved from the database. The authors state that the two methods compliment each other, as the manually built database improves precision of the system, and the MT system ensures recall for the items not present in the database (Ehrmann et al., 2011, p. 121).

Multilingual resources

Apart from named entity databases, other language resources can also be used in projecting annotations. Parallel corpora can be helpful resources, especially if they are aligned. Aligned parallel corpora contain annotations which map the units with the same meaning in the source and in the target text to each other. This kind of corpora may not exist or be readily available for all language pairs, but aligned parallel corpora can also be automatically constructed, without the need for excessive human labor.

Hou et al. (2019) takes advantage of a weakly chronologically aligned corpus in order to create projected annotations for Finnish. The authors used English as source and Finnish as target language, and collected news data for both languages. They then took capitalized words from the Finnish texts and marked them as 'named entity candidates', and did the same for named entities found by an English NER system in the source texts. The named entity candidates of both texts were then stemmed. The candidates in a Finnish article, published on a given date, were then matched to the named entities occurring in English articles published in +-2 days. The system worked fairly well for international names, but names not occurring in English texts (Finnish person names, place names) or names that are translated in Finnish ('France' -> 'Ranska') had to be dealt with using manually constructed word lists. However, they were able to build fairly well performing neural network models on the projected data.

Wikipedia is another resource often used in projecting named entities, used for example by Rijhwani et al. (2020). They used Wikipedia to retrieve NE candidates for the target language, along with probabilities denoting how likely the candidate is to refer to the same entity. These 'soft gazetteer features' with values between 0 and 1 were then used as features in the model, alongside word embeddings and character representations (Rijhwani et al., 2020). Rijhwani et al. (2020) used several low-resource languages (namely Kinyarwanda, Oromo, Sinhala, and Tigrinya) as target languages, and reported significant improvements gained by using these gazetteer features.

Candidate matching and model building

The various kinds of data transfer methods typically produce a list of named entity candidates for the target language. These may be for example top ranking machine translation results as well as possible database entries (Ehrmann et al., 2011), or various possible stems for the projected names (Hou et al., 2019). After retrieving the candidates, additional methods are needed to map them to the correct entities in the target language. In (Ehrmann et al., 2011), the correct candidate is chosen by applying three annotation projection methods in the order of strictness. These are exact string matching, consonant signature matching which allows some known variation in the consonants, and ignores the vowels, and finally edit distance.

After the named entities in the target text have been annotated with the projected annotations, what remains is to train a model with the resulting annotated data. Before the emergence of pre-trained contextual embeddings such as multi-lingual BERT, state-of-the-art results in NER were usually achieved by using a bi-directional LSTM, often combined with a Conditional Random Fields (CRF) layer. This was the model architecture used in Hou et al. (2019) as well as Rijhwani et al. (2020), although the latter also used an additional Convolutional Neural Network (CNN) layer.

Data transfer based methods can be a good option, if resources such as (aligned) parallel corpora, MT systems or bilingual name dictionaries are available. It works especially well if the source and target language are related to each other, or linguistically similar; this makes the projection easier. If other linguistic information is available in the target language corpus, such as part-of-speech tags and morphological information, they can be used to aid the performance of the model. The transfer process can therefore be tailored according to the specific language pair and resources available, which can yield good results especially for some target languages which massively multilingual systems perform badly on.

On the other hand, many approaches discussed above are somewhat labor-intensive and consist of several stages. In virtually all of the approaches, some kind of linguistic resources are required, which may not be available in low-resource languages. Depending on the type of projecting method used, many approaches also depend on the quality of the source language NER system. Finally, data transfer systems don't typically support zero-shot learning, i.e. classifying instances in a language that did not occur in the training data.

2.4 Model transfer methods

Multi-lingual BERT

Model transfer based approaches aim to produce a model that is inherently multilingual; it has learnt language-independent features from multilingual data, and can therefore be used to classify texts in several languages. Most common model transfer approaches in NER build on multilingual embeddings, often using pre-trained multi-lingual BERT (Devlin et al., 2018) or other similar transformer-based approaches.

Moon et al. (2019) demonstrate two crucial results of using BERT for NER: firstly, they achieved better results with the multilingual BERT model than with monolingual models that were trained for each language individually. Secondly, they achieved state-of-the-art results for three languages in the CoNLL data set (Tjong Kim Sang, 2002) and (Tjong Kim Sang and De Meulder, 2003) in zero-shot settings, using their multilingual model for languages that the model had not seen during fine-tuning. For the zero-shot experiment, they used 3 of the 4 CoNLL languages to fine-tune the BERT model, and tested its performance on the fourth, unseen language. They did the same for another dataset, OntoNotes 5.0 (Weischedel et al., 2011) They built two multilingual models, one with all 4 ConLL and one with all three OntoNotes languages, and compared them to monolingual models trained with only the target language, as well as a model that was trained only with English. In both the zero-shot and the multi-lingual setting, their models performed very well compared to previous results on the same data sets, achieving state-of-the-art results for several languages. There was variation between the languages, though; the model performed poorly for Arabic in zero-shot setting, which the authors assume was due to the WordPiece representations for Arabic not being optimal.

Moon et al. (2019) also reported a few techniques for improving the model's performance in zero-shot settings. Curiously, the F1-score tended to decrease during epochs, which the authors hypothesize may be due to the original BERT embeddings being well-aligned (Moon et al., 2019, p.4). Freezing some of the lower layers of the transformer generally helped with the issue. They also noticed that turning the task into a multi-task learning setting, adding secondary tasks such as language identification and cloze task, helped preventing the decrease of the F1-score over epochs.

Deeper look into multi-lingual BERT

Pires et al. (2019) conducted similar experiments with multi-lingual BERT, focusing on testing how it performs in zero-shot settings in two tasks: named entity recognition and part-of-speech tagging. Using the CoNLL data set, as well as an in-house data set of 16 languages, they achieved similarly good results as Moon et al. (2019). Interestingly, they also conducted experiments to determine where multilingual BERT's ability to create multilingual representations stems from. One question they posed was how much lexical overlap does play a role. They tested this in a zero-shot setting, using two types of BERT embeddings: monolingual English embeddings (EN-BERT), and multilingual (M-BERT). Pires et al. (2019) then plotted the model's F1-score versus the ratio of shared WordPieces of the languages.

They found out that the performance of the EN-BERT model relied almost solely on the shared WordPieces, achieving F1-scores close to zero when no WordPieces were shared, such as when the languages were written with different scripts. However, M-BERT achieved F1-scores between 40% and 70% even when the ratio of shared WordPieces was almost zero (Pires et al., 2019, p.3). Therefore, they concluded that M-BERT's ability to generalize across languages is not based on word similarity. However, comparing the performance to the number of shared WALS features showed that M-BERT's performance improves when the languages have linguistically similar structures. As for why M-BERT succeeds to generalize across languages, Pires et al. (2019) conclude that they believe that having WordPieces shared by all languages (such as numbers and URL's) forces the model to also map the other co-occurring pieces to a shared multi-lingual space.

Other approaches

Apart from off-the-shelf multilingual pre-trained models, more specialized types of model transfer have been proposed, some of which take advantage of similarity of a specific language pair. Murthy et al. (2018) attempted to improve NER tagging performance in low-resource settings, using several Indian low-resource languages as target languages, and a linguistically related Indian higher-resource language (namely Hindi) as an assisting language. They trained CNN-Bi-LSTM models for each low-resource language, using the small amount of annotated training data they had of the target language, and adding training data from the assisting language (with a smaller weight). They compared two kinds of models: ones which shared only sub-word features between the languages, as well as models sharing all training data and features between the languages. For all the low-resource languages except one, the completely multilingual model performed better; in the fourth language, the performance was similar to the sub-word model. However, negative effects of the multilingual training were also reported, such as the model getting confused by the multilingual data in cases where a token had a different meaning in different languages (Murthy et al., 2018, p.15). Finally, this approach seemed to only be useful in settings where the target language and assisting language are closely related.

Moon et al. (2019) list several of the benefits of a single multi-lingual NER model compared to several monolingual ones: it is easier to deploy and maintain, it scales well, it has the same memory/CPU/GPU footprint, and has the ability to be used in a zero-shot context (Moon et al., 2019, p.1). However, their observation of the poor performance of the model in Arabic in zero-shot setting indicates that there are clear differences in performance between languages. Another problem is brought up by Chen et al. (2019). They state that in a multi-lingual model transfer setting, a model usually cannot take advantage of the fact that the source language can be more similar to some of the target languages than others. For example, if transferring from English, Spanish and Chinese to German, the similarity of English and German is ignored, and the model cannot utilize the similarities between the two languages fully. English has as much weight as Chinese, a linguistically very different language, in the training (Chen et al., 2019, p.2). The result is that only aspects shared by all three source languages are learnt by the model.

2.5 Other types of approaches to multi-lingual NER

Multilingual NER systems, which don't directly fall under either data transfer or model transfer, include for example systems where several (often monolingual) models are built. The same test set can then be evaluated on many models, and one can attempt to determine the correct result based on the outputs of several models. This can be conducted for example by voting, where the result suggested by the biggest number of models is chosen as the correct result.

A more sophisticated approach to the problem is presented in Rahimi et al. (2019). They build several monolingual models, and use a Bayesian model to determine which model gives reliable results for each instance in a multilingual testing data. Their approach is based on the insight that a poor model makes diverse mistakes, but the few well-performing models provide consistently reliable results (Rahimi et al., 2019, p. 1). For this, they used the Bayesian model, which took each model's confusion matrix as an input and then evaluated the reliability of the model's results. This approach yielded strong results, and provided a way to use similarity between languages without explicitly telling the model which language each testing sample is in. It also clearly outperformed simple voting-based methods.

2.6 Comparing and combining model transfer and data transfer

In my thesis, I implement both a data transfer approach and a model transfer approach, and attempt a third approach which combines both data and model transfer. I implement all three approaches for four languages (Dutch, Estonian, Finnish, and Spanish), and compare their performance. I chose to work with these four languages, as Dutch and Spanish are included in the CoNLL NER data sets, and it is therefore easy to find comparable results for the test sets from previous results. On the other hand, Finnish and Estonian represent languages where there are less annotated data available, and they can be considered low-resource languages compared to the others. In order to follow a multilingual implementation, which would have the potential to work also on smaller languages not included in the experiments, I attempted to choose methods which require as few linguistic resources as possible.

1. Building models with projected annotations as training data

This approach represents data transfer, and is based on building models using projected annotations as training data. I use an existing parallel corpus, which contains sentence and word alignments for each of the four language pairs (namely English as source language and Dutch, Estonian, Finnish, and Spanish as target languages). The English source text for each language pair is then annotated, using an off-the-shelf English BERT model.

Once the source text has been annotated for named entities, the sentence and word alignments are used to map the named entities in the source text to the named entities in the target text. The resulting data sets containing projected annotations are used to train a mono-lingual bi-directional LSTM for each of the target languages, and each model is evaluated on an existing annotated test set.

No other linguistic resources are needed for this approach except for the aligned parallel corpus. Figure 2 shows the projection approach.

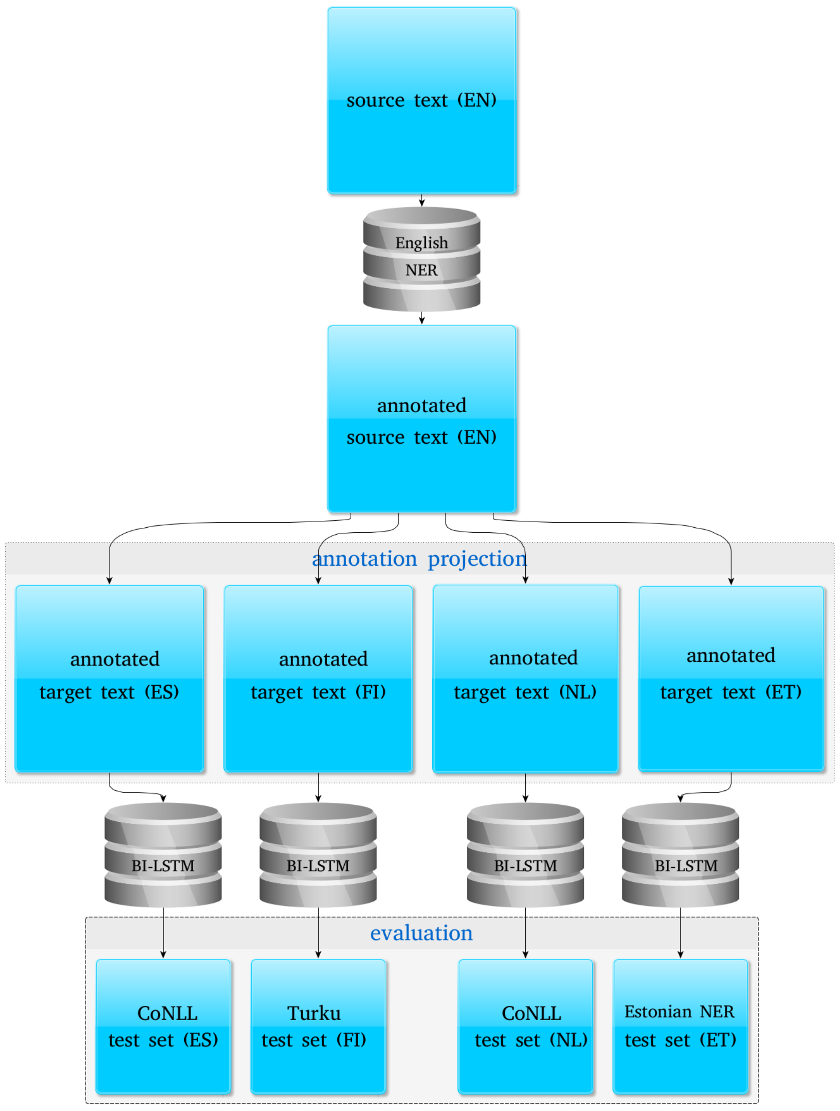

source text (EN)
English NER
annotated source text (EN)
annotation projection
annotated target text (ES)
annotated target text (FI)
annotated target text (NL)
annotated target text (ET)
BI-LSTM
BI-LSTM
BI-LSTM
BI-LSTM
evaluation
CoNLL test set (ES)
Turku test set (FI)
CoNLL test set (NL)
Estonian NER test set (ET)

Figure 2: The data transfer approach

2. Multilingual BERT

The second approach represents model transfer, and is similar to that presented by Moon et al. (2019). Multilingual BERT is used, and it is fine-tuned with a multi-lingual training data set of annotated NER data, created by mixing together the training data available for the four languages. This multi-lingual model is then evaluated on each of the training sets for the four languages. Even though annotated fine-tuning data is needed, this approach can be applied to low-resource languages as well, since fairly good results can be achieved with a relatively small amount of annotated data.

3. Multilingual BERT with projected annotations as fine-tuning data

The third approach implemented in the thesis attempts to combine data and model transfer. I simply utilize the projected annotations produced in approach 1, and fine-tune the multilingual BERT with that data. If successful, this will provide a way to use multilingual BERT without pre-existing annotated data. Figure 3 the BERT pipeline, which is used in both approach two, using annotated data for fine-tuning, as well as for approach three, using projected data.

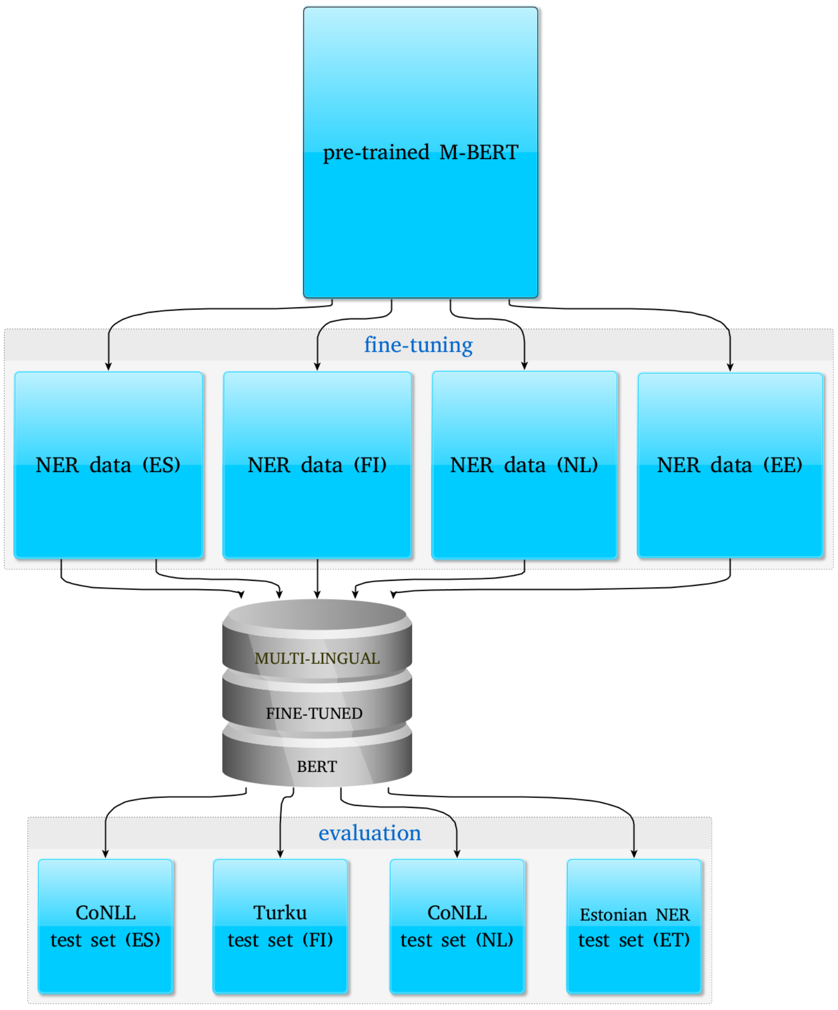


Figure 3: The model transfer approach

3 Data

3.1 Annotated NER corpora

In order to have annotated test sets for testing my approaches and for comparing their performances to earlier experiments, I used an annotated NER data set for each of the four target languages covered in the thesis. CoNLL data sets were chosen for their wide usage as bench marking sets for Dutch and Spanish. For Finnish, I chose the corpus by Luoma et al. (2020), as it is newer and covers a wider range of genres than Ruokolainen et al. (2019), which consists of technology news. For Estonian, I chose the only annotated NER corpus available.

CoNLL 2002 and CoNLL 2003 Named Entity Corpora

The CoNLL 2002 (Tjong Kim Sang, 2002) and CoNLL 2003 (Tjong Kim Sang and De Meulder, 2003) corpora are widely used as a benchmark in evaluating NER systems. The 2002 CoNLL Named Entity Recognition task included Dutch and Spanish, and the 2003 task included English and German. The annotation scheme is a variant of the IOB scheme, and it consists of four types of entities: persons (PER), organizations (ORG), locations (LOC) as well as miscellaneous names (MISC) (Tjong Kim Sang, 2002). For each four tags, there are B-tags and I-tags. B-tags denote that the word is the first word in a named entity of that type, and all other words that belong to the same entity are marked with I-tags. Words that are not a part of named entities are marked with 'O'. This results in a total of 11 unique tags.

For each language, the CoNLL corpora consist of a training set, a development set and a test set. Number of sentences and tokens in each each set for Dutch, Spanish and English is represented in Table 1.

Table 1: Number of sentences and tokens in CoNLL corpora

	train		dev		test	
	sentences	**tokens**	**sentences**	**tokens**	**sentences**	**tokens**
Dutch	15,806	218,737	5,195	40,656	2,895	74,189
Spanish	8,323	273,037	1,517	54,837	1,915	53,049
English	14,987	203,621	3,466	51,362	3,684	46,435

The data consists of news texts. The Dutch data is from a Belgian newspaper 'De Morgen', and the Spanish data consists of news wire articles from the Spanish EFE News Agency (Tjong Kim Sang, 2002).

Turku NER corpus

Luoma et al. (2020) compiled a new annotated Finnish named entity corpus. The corpus consists of texts extracted from the Universal Dependencies Finnish Treebank, and covers several genres, topics and writing styles (Luoma et al., 2020, p.1). Sizes of the training, development and test sets are presented in Table 2.

Table 2: Number of sentences and tokens in Turku NER corpus

	sentences	**tokens**
Train	12,217	162,746
Dev	1,364	18,308
Test	1,555	21,062

The annotation scheme of the corpus follows the IOB scheme, but differs from the CoNLL scheme in the number of entity types. 'MISC' type is not included, but instead there are separate types for products, events and dates. In order to use this data set for evaluation, I needed to convert the annotation scheme to correspond to the CoNLL scheme. Luoma et al. (2020) state that their product and event types are labeled as 'MISC' in CoNLL (Luoma et al., 2020, p. 3), so I converted them to 'MISC' tags. Dates are not annotated as named entities in the CoNLL corpora, so I converted all date tags to O-tags.

Estonian NER corpus

The Estonian NER corpus (Tkachenko et al., 2013) consists of news stories published in Estonian online newspapers 'Delfi' and 'Postimees'. The news stories cover a wide range of topics, including politics, economy and sports. The total size of the corpus is 14,859 sentences and 217,224 tokens. The corpus contained no splits into training, development and test sets, so I split the data myself. The sentences were mixed randomly, and then the data was split roughly 80/10/10 to training, development and test sets. The number of sentences and tokens in each set is represented in Table 3.

In this corpus, the annotation scheme consists of only three types of named entities: person, location and organization. There is no 'MISC' category. This was taken into account by removing the 'MISC' tags from the projected training data, thus having the same annotation scheme in the training data as in the test set when training the bi-LSTM. For this reason, the results for Estonian are not directly comparable to those of the other languages, as the Estonian data has fewer categories and is therefore easier for the models to classify.

Table 3: Number of sentences and tokens in Estonian NER corpus

	sentences	**tokens**
Train	12,058	175,714
Dev	1,399	20,601
Test	1,400	20,909
Total	14,859	217,224

3.2 Unannotated data

Tilde MODEL Corpus - Multilingual Open Data for European Languages

To build a model on projected annotations, my approach required a parallel corpus that would be aligned both at sentence level and at word level. For this purpose, I decided to pick a corpus from the OPUS collection of corpora (Tiedemann, 2012), which contains several corpora of translated texts. Most of the corpora have been automatically aligned both at sentence level and at word level. I chose to work with the Tilde MODEL Corpus (Rozis and Skadiņš, 2017), since it supported all four of my language pairs (EN-NL, EN-ES, EN-FI, EN-ET). It also contained word alignments and covered the widest number of genres of the corpora available. None of the corpora on OPUS which contained the desired language pairs contained news texts, which would have been the optimal genre, considering that the CoNLL corpora as well as the Estonian annotated corpus consist of news texts.

The Tilde MODEL corpus consists mainly of European Commission press releases, European Medicines Agency documents, European Economic and Social Committee documents, and texts from the European Central Bank web site as well as the web site of World Bank (Rozis and Skadiņš, 2017, p. 2). Also, smaller websites were crawled to include texts from the domains of culture and travel. These are included for some of the four language pairs but not all. As the majority of the texts come from the European Union, the quality of the translations is high, which is important when projecting the annotations from one language to another.

Rozis and Skadiņš (2017) note that the European Commission press releases' contents are translated exactly, so they are sure to contain the same named entities in source and target sentence. However, most of the texts included in the Tilde MODEL coprus are fairly specialized, and cannot be considered general language, which may pose problems when the models are evaluated on the annotated sets, that contain quite different kind of data.

To download the corpus in a sentence-aligned format for each of the four language pairs, I used OpusTools (Aulamo et al., 2020). OpusTools provides a command to extract aligned lines, and a file is provided for aligning the sentences at word-level. For each sentence, the file provides the indices of corresponding words in the source and target sentence. The word-alignments have been created automatically, as well as the sentence alignments. As no manual corrections have been carried out, there may be mistakes in the alignments at both sentence and word level. Number of aligned sentences extracted for each language pair is represented in Table 4.

Table 4: Number of parallel sentences extracted from Tilde MODEL corpus

language pair	**sentences**
EN-NL	3,519,555
EN-ES	3,760,796
EN-ET	2,059,556
EN-FI	3,053,839

4 Methods

4.1 BERT models

BERT and multilingual BERT

BERT stands for 'Bidirectional Encoder Representations from Transformers', and it was introduced in 2018 by Devlin et al. (2018). BERT is a multi-layer bidirectional Transformer encoder, consisting of a stack of encoders. The model architecture is based on the Transformer, introduced in Vaswani et al. (2017). BERT models are pre-trained on two unsupervised tasks, a masked language model task and a next sentence prediction task (Devlin et al., 2018). The purpose of pre-training on a huge corpus is to teach the model general information about language, so it will have the ability to solve many kinds of Natural Language processing tasks.

Two pre-trained English BERT models were released: BERT-base, and BERT-large. BERT-base is a reasonably sized model, with 12 encoder layers, 768 hidden layers, 12 attention heads, and 110M parameters. BERT-large is a huge model, with 24 encoder layers, 1024 hidden layers, 16 attention heads and 340M parameters. BERT models use WordPiece tokenization, developed by Wu et al. (2016). WordPiece tokenization is a data-driven tokenization method, intended for reducing the number of out-of-vocabulary words. Frequent words stay as they are, but rarer words are split into smaller units. In this way a balance between flexibility of characters, as well as the usefulness of words as units, is achieved (Wu et al., 2016, p. 7).

The chosen pre-trained BERT model can then be fine-tuned, using annotated training data, to optimize it for the desired NLP task. Devlin et al. (2018) reported that the fine-tuned models achieved state-of-the-art results for 11 Natural Language Processing tasks. Fine-tuning the BERT model is computationally quite inexpensive, and the same pre-trained model can be used for many NLP tasks, without task-specific modifications to the architecture.

Named Entity Recognition was among the tasks where results competitive with the state-of-the-art were achieved. Devlin et al. (2018) tested the system in two ways: firstly, they used the normal fine-tuning method and fine-tuned BERT with the CoNLL English data. This resulted in an F1-score of 92.8 with BERT-base, and 92.8 with BERT-large. Secondly, they tried the feature-based approach, where no fine-tuning is carried out. Here, they took activations from one or several layers of BERT-base, and used those as input to a bi-LSTM. This approach also yielded results comparable to the state-of-the-art.

Devlin et al. (2018) released both and English and a multilingual BERT. The English BERT model was pre-trained on the BooksCorpus (800M words) as well as on English Wikipedia (2,500M words). The multilingual model was pre-trained on the top 100 languages with the largest Wikipedias, and the whole Wikipedia dump (excluding user and talk pages) was used for every language.

Building the BERT models

The pre-trained multilingual BERT model is available in the HuggingFace library.[1] I used it to build two models: one fine-tuned with annotated data, and one with projected data. Both models were fine-tuned with multilingual data, which was a combination of all four training sets of all four languages. The data was mixed randomly. A multi-lingual development set, created in the same manner, was also used.

Annotated BERT model

To train a multi-lingual BERT model with annotated data, I used the transformers library by HuggingFace [2]. This experiment conducted in a similar manner to that implemented in Moon et al. (2019), where the writers took annotated data of all of the CoNLL languages (ES, EN, DE and NL), and used m-BERT to fine-tune a multilingual model. They mixed the training data of all 4 languages together randomly, and used that for training.

I did an experiment which was otherwise similar, but partly with different languages. I used the annotated CoNLL sets of Spanish and Dutch, as well as annotated data of Finnish and Estonian. In Moon et al. (2019), the F1-scores they achieved with the method were quite high, outperforming monolingual BERT models. The pre-trained BERT model used in the article and in my experiment was bert-base-multilingual-cased.

I anticipated to get a relatively well performing model as well, but my hypothesis was that the introduction of Finnish and Estonian could lower the results, as they belong to the Finno-Ugric language family and are therefore linguistically distant from the others. The four CoNLL languages are all Indo-European, and 3 of them belong to the Germanic language family.

One problem was caused by the fact that not all of the data sets I used had the same named entity tags in them. As explained in the data section, the Finnish set was converted to contain the same tags than the CoNLL sets, but the Estonian was missing all the 'MISC' tags. I was not sure whether it will be a problem, but I proceeded to fine-tune the BERT with all four languages even despite the non-matching annotation schemes. As in the paper, I took all the training data I

[1]https://huggingface.co/bert-base-multilingual-cased

[2]https://github.com/huggingface/transformers

had in the languages, and mixed them randomly. Then I pre-processed the data to contain only the tokens and the named entity tags. Then I used a pre-processing script[3] to restrict the length of sentences to a maximum of 128 WordPiece tokens.

I used the bert-base-multilingual-cased BERT model. To fine-tune, I used a script provided in the transformers library, which is intended for fine-tuning named entity recognition models. More specifically, I used the older version of the script[4], as it could handle my file format. I changed the default train and test sets to my own, but kept all other parameters as the defaults. As in the pre-processing phase, maximum sequence length was set to 128 WordPiece tokens, and the model was trained for three epochs.

Projected BERT model

A similar model was trained using projected data for fine-tuning. All other aspects of the model and fine-tuning were the same as in the annotated BERT model described above, except for the data sets. The training and development sets consisted of TildeMODEL data with projected annotations.

4.2 Annotation projection models

The annotation projection approach I chose consists of three major stages:

1. Initial annotation of English source data
2. Projecting the annotations
3. Building bi-LSTM models trained on the annotations

In this section, I first will present the steps that were taken to annotate the English data, and then proceed to describe the models and how they were built. Finally, I will go through the process of annotation projection, and present results of preliminary experiments I did to test the best projection approach.

4.2.1 Initial annotation of English source data

Off-the-shelf BERT for NER for annotating parallel source data

To get initial NER annotations for the English source data, I used bert-base-NER, an off-the-shelf BERT model fine-tuned for English Named Entity Recognition tasks. It is available from

[3]https://raw.githubusercontent.com/stefan-it/fine-tuned-berts-seq/master/scripts/preprocess.py
[4]https://github.com/huggingface/transformers/blob/master/examples/legacy/token-classification/run_ner.py

Hugging Face.[5] This model was chosen to create the initial annotations, because BERT based systems represent the state of the art in NER, and because of BERT's ability to generalize.

This model was fine-tuned on the English CoNLL-2003 Named Entity Recognition data set, and it achieves near state-of-the-art results on the CoNLL test set. The model was trained using the same hyper-parameters as in the original BERT paper (Devlin et al., 2018), and it achieved an F1-score of 91.3. It is noted that the F1-score obtained in the original paper is slightly higher (92.4 with BERT-base). It is stated that this is due to Devlin et al. (2018) having experimented with CRF, and encoding document context in their experiments.

BERT's pretraining on a large corpus allows it to generalize, even though the fine-tuning is conducted on a fairly small dataset (about 15k sentences in this case). However, it is noted that the fine-tuning is conducted on a collection of news texts, and the training data only spans over a specific time. This will likely restrict the model's ability to generalize in other domains and contexts.

Mapping BERT tokens to words

Bert-base-NER, like BERT models in general, tokenize the input into WordPiece tokens, and the output contains an annotation for each WordPiece as well. When doing Named Entity Recognition, the desired output is one annotation per word, which means the BERT output must be mapped to word tokens. In addition to that, when attempting to project the annotations to another language using word alignments, special care must be put into preserving the original indexing in the output annotations.

This can be illustrated by the following example, using an English sentence from the EN-FI parallel corpus:

> A city where ballet was born under the genius of Petipa and the Russian Revolution kicked off , St. Petersburg can seem too much for even a lifelong city-break .

The tokens are separated from each other with a space, and the sentence consists of 30 tokens. This tokenization is important to preserve, as the target text is tokenized in the same way, and the word-alignments used in the projection step point to indices in the source and target sentence that are based on this tokenization.

For this sentence, the tokenization by the bert-base-NER, along with the output of the model, is represented here (only tokens annotated as names are marked here, other tokens have received tag 'O'):

> [CLS] A city where ballet was born under the genius of Pet$_{\text{B-PER}}$ ##ip$_{\text{I-PER}}$ ##a and the Russian$_{\text{B-MISC}}$ Revolution$_{\text{I-MISC}}$ kicked off , St$_{\text{B-LOC}}$.$_{\text{I-LOC}}$ Petersburg$_{\text{I-LOC}}$ can seem too much for even a lifelong city - break . [SEP]

[5]Available at https://huggingface.co/dslim/bert-base-NER

The length of the BERT output is 37 tokens, and it includes the special BERT tokens '[CLS]' and '[SEP]'. Furthermore, the WordPiece tokenizer has split the name 'Petipa' into three pieces, 'Pet', 'ip' and 'a'. These tokens have received annotations 'B-PER', 'I-PER' and 'O', respectively. The BERT tokenizer also treats punctuation marks differently, considering the full stop in 'St. Petersburg' and the dash in 'city-break' to be separate tokens.

Using an approach suggested in the BERT GitHub directory[6], I mapped the WordPiece tokens to the original tokenization. This was conducted by tokenizing each original token individually, and saving the indices of the corresponding tokens. In case the original token was split into several pieces, like in the above example, the annotation for the first WordPiece token was always used. This was done the same way in the NER experiments in Devlin et al. (2018) as well. This resulted in the following final annotation for the sentence:

> A city where ballet was born under the genius of Petipa$_{\text{B-PER}}$ and the Russian$_{\text{B-MISC}}$ Revolution$_{\text{B-MISC}}$ kicked off , St.$_{\text{B-LOC}}$ Petersburg$_{\text{I-LOC}}$ can seem too much for even a lifelong city-break .

Evaluating annotation quality

As the quality of the projected training data is directly dependent on the quality of the original named entity annotations, a small sample of the annotated English source data was manually evaluated. A set of 302 sentences was chosen randomly from the data, of which 118 were considered to have named entities in them. The following were counted from the data:

1. Correctly identified names
2. Names identified as names, but with incorrect category (such as 'B-MISC' for 'B-PER')
3. Names not identified as names (marked with 'O' tag)
4. Non-names incorrectly identified as names

A more detailed annotation was not attempted, as it was considered too time-consuming, and calculating precision and recall for every individual class would not be very informative, given such a small dataset. Additionally, it is not entirely clear what should be included in the 'MISC' category - it contains at least events, and adjectives derived from country names, but a more detailed explanation was not available. In some cases, the data contained abbreviations which seemed to refer to named entities such as organizations, but it was not possible to determine what entity they referred to based on the text. In such cases, decisions on whether they were names or not, and whether the category assigned to them was correct, were made solely based

[6]Available at https://github.com/google-research/bert

on the context; if it seemed plausible it was a name, it was marked as a name. No external information was searched to determine what kind of an entity was referred to. For these reasons, the manual evaluation may not be a hundred percent accurate; however, unclear cases represented only a small part of the cases.

A confusion matrix for the manual annotations is pictured in Table 5.

Table 5: Named entity annotations by bert-base-NER compared to manual evaluation

		True label		
		Name	Not name	Total
Predicted label	Name	319	91	410
	Not name	27	6353	6380
	Total	346	6444	6790

Here, the 27 instances of actual names misclassified include both type 2: names identified as names but with wrong category (12 instances), as well as type 3: names not recognized as names (15 instances). This results in the following scores for the evaluation:

Metric	**Score**
precision of names	0.778
recall of names	0.922
F1-score of names	0.844
accuracy	0.983

Considering that the projected data set is not in the same domain that the BERT model was fine-tuned with, the results are satisfactory. Recall of names is especially important, as it determines how many correct instances there can be in the training data. As for precision, some of the mistakes will be fixed in the following projection step, where for example incomplete tag sequences are taken into account.

Mistakes in the annotation were most typically of the two following types: capitalized words, which were not names, but that were incorrectly identified as names, and incomplete tag sequences, where the beginning of the named entity was recognized, but not the latter part. As an example of the first type, the word ‘Member’ is wrongly classified as a location in the following sentence:

> a subsidiary of the parent undertaking of an investment firm or credit institution authorised in another Member$_{\text{B-LOC}}$ State ;

An example of the second type, where ‘97/70/EC’ should also be included in the preceding named entity, and thus marked with ‘I-MISC’:

Council$_{\text{B-MISC}}$ Directive$_{\text{I-MISC}}$ 97/70/EC of 11 December 1997 setting up a harmonised safety regime for fishing vessels of 24 metres in length and over

4.2.2 LSTM trained on projected embeddings

Background

To build models using projected embeddings as data, neural network models were chosen as the implementation. As influential results such as Lample et al. (2016) and Ma and Hovy (2016) show, before BERT, the state-of-the-art in NER was neural network models, more specifically bi-directional Long Short-Term Memory models (bi-LSTM's). Best results have usually been reached by models using both character and word level representations, and a Conditional Random Fields (CRF) layer. Sometimes, such as in Lample et al. (2016), also a Conditional Neural Network (CNN) layer is used.

As Ruokolainen et al. (2019) note, the strength of these approaches stems from their use of both character-based word representation models, as well as distributional representations of the words, namely pre-trained embeddings. These representations are then combined, and fed into the bi-LSTM neural network. Having a bi-directional model is important in sequence labeling tasks, as the model goes through the input from both directions, which helps in detecting patterns in the sequences of tags. On top of the LSTM, there is a CRF layer, which as Ruokolainen et al. (2019) state helps to recognize the dependencies between adjacent tags (Ruokolainen et al., 2019, p.22). For example, an I-tag cannot be preceded by an O-tag. This kind of architectures have been shown to yield strong results, without the need for hand-crafted features or gazetteers.

Implementation

For building the neural network, I used the NCRF++[7] toolkit for neural sequence labeling (Yang and Zhang, 2018). It provides an implementation of a similar model as Lample et al. (2016) and Ma and Hovy (2016). The toolkit can be configured to fit to many kinds of sequence labeling tasks, and all parameters are easily configurable. When using the default configuration, NCRF++ builds a Char CNN + Word biLSTM + CRF model, using pretrained Glove embeddings for the words.

To build four models, one for each target language, I used the default parameters. This meant using a SGD optimizer, 100 epochs, a batch size of 10 and a dropout of 0.5. The learning rate was set to 0.015 and training was conducted on CSC's Puhti supercomputer using a GPU. I chose to use FastText pretrained embeddings (Grave et al., 2018)[8] for the initial word representations for Spanish, Dutch and Estonian. The file contained word embeddings for 2

[7]Available at https://huggingface.co/dslim/bert-base-NER
[8]Available at https://fasttext.cc/docs/en/crawl-vectors.html

million words from each language, and the length of the vectors is 300. The vectors were trained on the Common Crawl data as well as Wikipedia. For Finnish, I conducted some initial tests of building models from annotated data, and tried several word embedding files. Using the Finnish Internet Parsebank word2vec embeddings (Kanerva et al., 2014) gave the best results, so it was also utilized for the main experiments. The word embedding file contains embeddings for 4,2 million words, and the length of the vectors is 300 as well.

Data used in the final experiments

For each language, I used one tenth of the available data I had, which meant about 200k - 300k sentences per language. That was split 90/10 into a training set and a development set. 6 shows the number of sentences and tokens in the sets for each language.

Table 6: Number of sentences and tokens in projected annotations training and development sets

	train		dev	
	sentences	**tokens**	**sentences**	**tokens**
Dutch	308,988	7,247,122	34,353	800,547
Spanish	338,012	9,038,629	37,545	1,003,853
Finnish	274,176	4,565,332	30,467	503,364
Estonian	185,059	3,142,133	20,557	349,042

4.2.3 Projecting annotations

In annotation projection, annotations in one language are projected on the data of another language. As described in the background section, this can be done in several ways. Here, my approach was based on using the word-alignment files in the corpus. In the annotation projection step, I took the annotated English source sentences produced in the previous step, as well as the corresponding target language sentences. Additionally, I took the word alignment file, which maps the words in the source language to the words in the target language that have the same meaning.

An example from the EN-FI corpus is illustrated in Figure 4. It displays the sentence in the source and target languages, each token represented in its own square. The arrows indicate the word alignment links between the tokens. The yellow tags indicate the annotations for the source sentence, and the resulting annotations for the target sentence after projection.

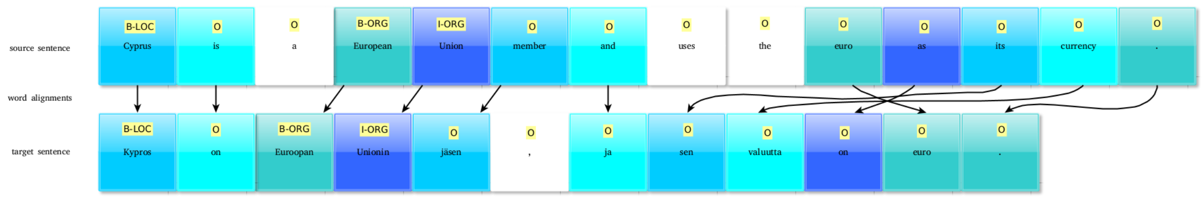


Figure 4: Example of a word aligned sentence in the corpus

Here, the word ‘Cyprus’ has been marked with ‘B-LOC’ tag in English, and the alignments map that word to the corresponding Finnish word ‘Kypros’ in the Finnish sentence. ’European Union’ is mapped to ’Euroopan Unioni’, and annotated as ’B-ORG I-ORG’. The rest of the tokens have the annotation ’O’, denoting that they are not a part of any named entity. Thus, projecting the annotations from English to Finnish as indicated by the arrows results in a correctly annotated sentence in Finnish.

In some cases, the word alignments have links from one source word to several target words, and vice versa. In cases, where one source word is linked to several target words, all of the target words get the same annotation as the single source word. This may result in mistakes, for example if the several target words occur next to each other, and are a part of the same named entity. In this case, all of the target words would receive a B-tag, and would be considered to belong to separate named entities. However, this kind of cases are not likely to occur very frequently in the data.

In cases where several source words link to one target word, the target word receives the annotation of the source word that has the biggest index, i.e. the one that occurs last in the sentence. This strategy was chosen out of convenience, as no reliable way of choosing the best annotation was available. Finally, in cases where there are words in the target sentence that have no links pointing at them, the words receive an ’O’-tag.

Annotation projection can be a relatively easy way to produce training data, but mistakes in the projected annotations can arise from various sources:

1. Sentence alignments
2. Word alignments
3. Source data annotations
4. Projection step

Mistakes of type 1 and 2 arise from the corpus itself, where sentence alignments and word alignments have been created automatically, and no manual corrections have been carried out. I did not attempt to correct incorrect sentence alignments. For the word alignments, several different ways to align the words was provided in the corpus, and I tested which one would be the most suitable for the task. Alignment mistakes may lead to various different kinds of erroneous tags, such as names tagged with ‘O’ tag, non-names tagged with name tags, or incomplete tag sequences.

Mistakes of type 3, related to the quality of the original NER annotations, were already described in the previous section. Finally, type 4 refers to mistakes introduced in the projection phase. For example, different word order can cause mistakes. If the English source data would contain the named entity ‘European Union’, and it would be correctly annotated as ‘B-ORG’ ‘I-ORG’, in the Spanish target text this named entity would be ‘Unión Europea’. As the order of the words is different, the projection would label the entity as ‘I-ORG’ ‘B-ORG’ in the Spanish text, which is an incorrect tag sequence.

Testing different projection approaches

In order to correct mistakes, I tried several ways of projecting the annotations for Finnish. Three different word alignment methods were used, and for each, two versions of projections were made: one without any fixes, and one with fixes applied. The resulting six projected data files were then used to train a bi-LSTM model, and the results were evaluated against the annotated Finnish test set.

Different word alignment files

I tested training models with three different kinds of word alignment files. In the word alignments, the words are first aligned to two directions: from the source sentence to the target sentence, and vice versa. This results in two word alignment files, which may contain different links between the words. These two alignments can then be combined in several ways to produce the final word alignment file; this process is called symmetrization in the literature on statistical machine translation[9]. The first approach I used in the tests was intersect, where only the links, which exist to both directions (i.e. from source to target and from target to source) are included. This word alignment method is expected to give a high precision and low recall, as weaker links, only present in one direction, are ignored.

The other two approaches used are grow-diag, and grow-diag-final. In these, rules are applied to add links between words if certain conditions are met. These approaches give a higher recall than intersect, but precision may be lower, as more uncertain links between the words are included as well.

In total, 10% of the data (305,384 sentences) were processed in the EN-FI Tildemodel corpus. For each of the word alignment files, as the projection was carried out, some broken sentences were filtered out. In some cases, the source sentence, the target sentence and/or the

[9]See for example Philipp Koehn’s book on the topic: http://www.statmt.org/book/

alignment file were empty. These sentences were filtered out. In this data, there were 5 empty source sentences and no empty target sentences. In the intersect and grow-diag, there were 736 cases with empty alignments; in the grow-diag-final-and, there were 95.

It was also checked, whether the target index or the source index was pointing to an index that was not present in the sentence. In this data, no cases like this were found. All of these cases were considered as noise or mistakes in the alignments, and the whole sentence was excluded from training. For the grow-diag and intersect, 304,643 sentences remained after removing the broken sentences. For grow-diag-final-and, 305,284 sentences remained.

Fixed vs. unfixed versions

For each of the three word alignment types, two versions of annotation projection were carried out: one with no fixes, and one with manual fixes applied to the projection. The first fix attempts to correct cases that were mentioned above, where an I-tag would otherwise end up preceding a B-tag in cases where the order of constituents is different between languages. If a B-tag was followed by an I-tag of the same type, and the word alignment indices pointed to two adjacent words in the target text, but in the opposite order, the tag sequence was reversed for the resulting annotations.

This fix was applied simultaneously to the projection, so it was conducted before the second fix. The fix corrected several mistakes, and it is unlikely that it would have introduced new mistakes. In the Finnish data, an example of a fix was the word pair ‘West Ireland’, which was tagged as B-LOC I-LOC. The Finnish counterpart was ’Irlannin länsipuoli’, which would have otherwise been misclassified as ’I-LOC B-LOC’, but the fix caused it to be correctly projected as ’B-LOC I-LOC’. As mentioned before, this kinds of fixes could be especially useful in languages such as Spanish, where the order of nouns and adjectives in noun phrases is often different than in English, but in some cases also proved useful for Finnish where the word order is relatively free.

The second fix was applied to the whole target tag sequence, after all the words in the sentence had been processed. The sequence was searched for tag combinations where an ‘O’-tag was followed by an ‘I’-tag. These cases are always mistakes, as a named entity tag sequence must be started with a ‘B’-tag. If the preceding word, tagged with an ‘O’-tag, started with a capital letter, it was changed to a ‘B’-tag of the same type. If this was not the case, it was checked whether the word itself, tagged with an ‘I’-tag, started with a capital letter. If yes, the tag was converted into a ‘B’-tag of the same type. If not, the tag was converted into an ‘O’-tag.

This fix was not applied recursively, i.e. if for example converting the ‘I’-tag to and ‘O’-tag lead to another O -> I sequence, this was not fixed. Recursion might have caused the fix to have unforeseen consequences in more complicated tag sequences. This fix seemed to improve both the precision and the recall of the projected annotations. However, it also introduced some new mistakes to the projected data.

The fixes were applied to all three word alignment types. There were some differences to how many times each fix needed to be applied in each of them, shown in Table 7. I-B to B-I refers to swapping the order of two tags due to word order, and the rest of the columns contain counts for the three possible fixes applied to sequence-initial 'I'-tags. For grow-diag and grow-diag-final-and files, which are quite similar word alignment approaches, the number of fixes is fairly similar. Intersect file has less links between words, so it is natural that less fixes were applied to it.

Table 7: Number of fixes applied to each word alignment approach

file	**I-B to B-I**	**I to B-I**	**I to B**	**I to O**	**total**
intersect	350	1969	6507	19338	28164
grow-diag	523	1696	10244	32972	45435
grow-diag-final-and	532	1741	10572	35353	48198

Building test models using different projection strategies

In order to test which projection approach performed the best, I built six bi-LSTM models with the NCRF++ toolkit described above. This meant using the Finnish data described above, and using each of the three word alignment files to project the data. Additionally, each of the word alignment files was applied together with the manual fixes to produce three more data sets. Six models were then built. The architecture and parameters of the models was the same than I used in the final experiments. The models were trained models for 24 hours. This meant 56 epochs for intersect + fixes, 57 for grow-diag-final as well as grow-diag + fixes, 70 for grow-diag-final-and + fixes, 72 for grow-diag and 73 for intersect.

The training set and the development set consisted of projected data. The 300k sentences of data was split 90/10 to a training and development set. The test set was annotated data, the Turku NER corpus test set. Details of the data set (before splitting), such as the number of each kinds of named entity tokens, for each of the 6 data sets are listed in Table 8. The bold type marks the larger figure between the fixed and unfixed versions for all three strategies. The table shows that the fixes tended to increase the number of B-tags, and decrease the number of I-tags in all the files. The biggest impact was on the categories 'I-MISC', where the number of tags dropped significantly between the unfixed and fixed versions for all three files, and 'I-LOC', where the drop is especially drastic for grow-diag and grow-diag-final-and data.

Results of tests

F1-scores during training for the development and test sets are presented in Figure 5. Final results for the Turku NER test set for the models trained with the different projected data sets are presented in Table 9. For this table, the results for the epoch with the best performance on the development set were chosen.

As can be seen from the plots illustrating model performance between epochs, the results show a clear decrease along epochs for all the test models. For the development set, the results improve until around the tenth epoch, but after that there is a slow downward slope in the F1-scores as training continues[10]. For the test set, the best F1-scores are generally achieved within the first ten epochs, and the downward slope of the results is more drastic than for the development set.

Table 9 shows that there are no clear differences between the performance of the three word alignment approaches. The F1-scores range from 51 to 53. Recall is better than precision for all the tests, precision ranging between 49 and 51, and recall between 52 and 59. Clearest differences between the tests are visible in the recall, where the worst performing model is the one using grow-diag word alignments with a recall of 52.46. The best recall, 58.87, is achieved with the intersect + fixes alignments. This is somewhat surprising, as intersect has less links between words than the other approaches, and would be expected to give a better precision instead of recall. Intersect without fixes and grow-diag + fixes approaches also yield a slightly better recall than the rest of the models, otherwise differences between the different approaches are small.

As for the fixed versions, they seem to bring a slight improvement to the models, but the difference in the results is small. The fixes have the biggest effect on the grow-diag alignments, where the recall improves from 52.46 to 57.64. Also for intersect, the recall improves by two percentage points.

Based on these tests, I chose to continue with grow-diag + fixes approach for the word alignments, and to project the annotations for all the languages with that. Even though the differences between the models were small, the grow-diag + fixes test gave the highest accuracy and F1-score. The fixes did not provide clear improvements in the results in general, but they showed a consistent improvement to the results in most of the models, so they were decided to be included in the final experiments.

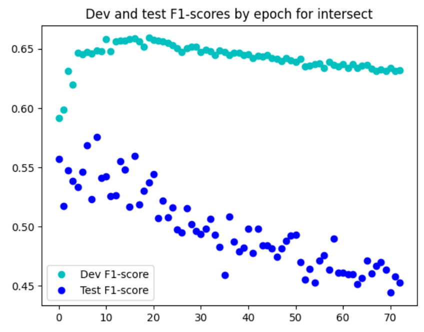
[10]This is also the case for the grow-diag-final-and test model, which may appear to behave differently in the plot. This is due to some very low F1-scores in the first epoch, which stretch the scale of the plot to be wider

Table 8: Number of tokens and sentences in projected data files

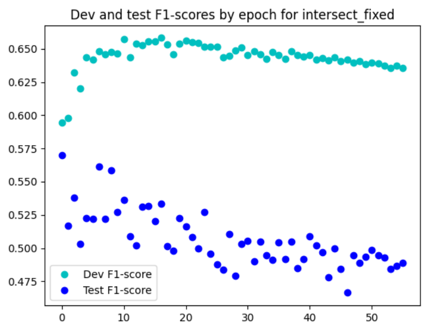
	intersect	intersect + f	grow-diag	grow-diag + f	final-and	final-and + f
sentences	304,643	304,643	304,643	304,643	305,284	305,284
tokens	5,373,339	5,373,339	5,373,339	5,373,339	5,375,647	5,375,647
total name	378,641	361,272	407,644	376,368	423,511	389,899
B-MISC	85,430	**89,483**	87,522	**92,220**	90,393	**95,231**
I-MISC	**43,737**	28,896	**53,069**	35,784	**55,443**	37,039
B-PER	13,868	**14,270**	14,342	**14,789**	14,516	**14,981**
I-PER	**6,116**	5,782	**6,289**	5,870	**6,368**	5,917
B-LOC	45,191	**45,950**	34,783	**37,780**	35,528	**38,565**
I-LOC	**4,713**	3,450	**19,799**	5,058	**20,211**	5,165
B-ORG	129,277	**132,539**	134,977	**138,805**	141,486	**145,489**
I-ORG	**50,309**	40,902	**56,863**	46,062	**59,566**	47,512

Table 9: Results for different projected data files for Finnish

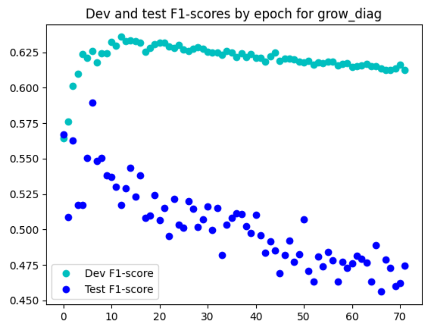
file	**accuracy**	**F1-score**	**precision**	**recall**
intersect	95.6	53.2	49.9	56.9
intersect + fixes	96.0	53.3	48.7	**58.9**
grow-diag	95.9	51.1	49.9	52.5
grow-diag + fixes	**96.1**	**53.3**	49.6	57.6
grow-diag-final-and	96.0	52.0	50.2	53.9
grow-diag-final-and + fixes	96.1	52.4	**51.1**	53.7



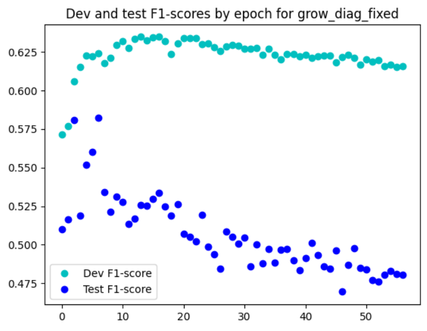
(a) Intersect



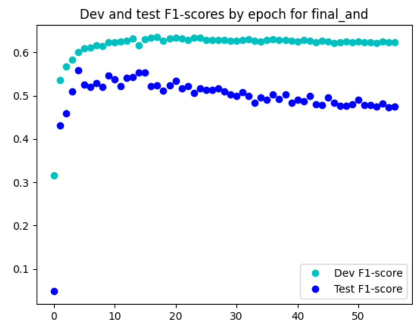
(b) Intersect with fixes



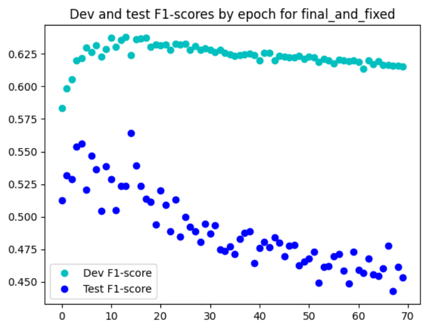
(c) grow-diag



(d) grow-diag with fixes



(e) grow-diag-final-and



(f) grow-diag-final-and with fixes

Figure 5: F1-scores of the 6 models over epochs for Finnish

4.3 Evaluation

In all the experiments, NER performance is evaluated in terms of exact mention-level. Precision, recall and F-score are calculated as implemented in the standard conlleval script[11], requiring both the type and span of predicted mentions to match a gold standard mention and summarizing the results as micro-averages. The seqeval framework (Nakayama, 2018) is used to achieve the same standard conlleval scores on Python, instead of the original Perl shell script.

[11]https://www.clips.uantwerpen.be/conll2000/chunking/conlleval.txt

5 Results

5.1 Annotation projection results

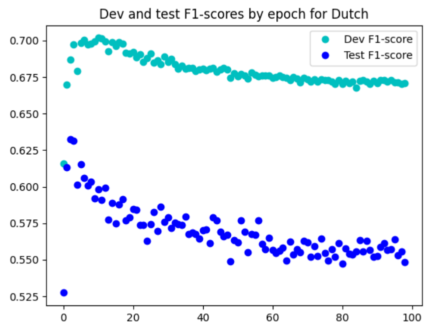
Based on the experiments conducted on Finnish, I chose to use the grow-diag word alignment file for all the languages, and to include the fixes made to the projection. I proceeded to train a model for each of the four languages. The model was the same as in the Finnish experiment, but now I trained each model for 48 hours. This meant 84 epochs for Spanish, 99 epochs for Dutch and 100 epochs for both Finnish and Estonian. Both the training set and the development set were projected data, and annotated test sets were used for all four languages (CoNLL sets for Spanish and Dutch, and Turku set for Finnish, and the Estonian annotated set). Sizes of the training and development sets were listed in the data section, in Table 6.

Table 10 shows how many fixes were done on each language data, combined for the training and development sets. As I hypothesized, the 'I-B to B-I' fix, which corrects mistakes introduced by a changing word order between the languages, occurred more in Spanish than in the other languages. Also the rest of the fixes had a bigger impact on the Spanish data set than in the other languages.

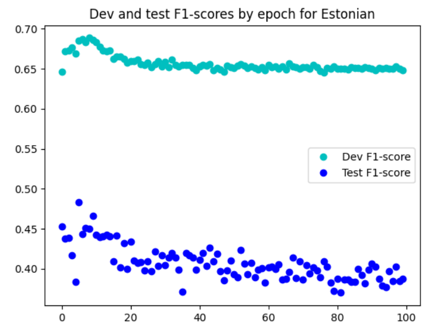
Table 10: Number of fixes applied to the projected data for each language

file	**I-B to B-I**	**I to B-I**	**I to B**	**I to O**	**total**
ES	8,988	6,137	63,251	41,458	119,834
ET	217	984	7,277	18,052	26,530
FI	523	1,696	10,244	32,972	45,435
NL	388	1,852	16,218	32,700	51,158

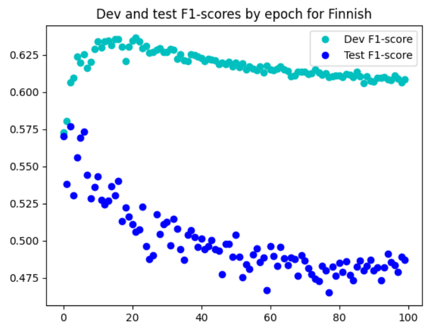
Performance of the models over epochs is pictured in Figure 6. The development of the F1-scores along epochs is similar to that reported earlier in the Finnish experiments: there is a downward slope in the results as training continues. This seems to suggest that the model is overfitting to the test data, and that explains the rapid decrease in the test set results. However, there is a slight decrease in the development set results as well, even though that consists of similar data than the training set. For Spanish, the plot shows a strange behavior for some epochs: the F1-score drops to 0.00 and -1 for the development and test sets. It is not clear what causes this, and the issue was already present in some preliminary experiments.



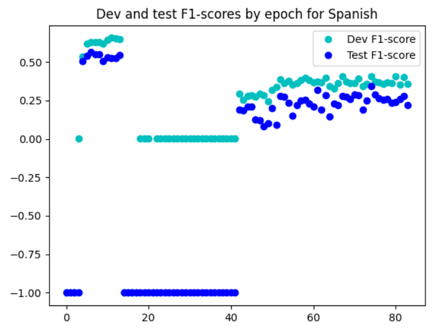
(a) Dutch



(b) Estonian



(c) Finnish



(d) Spanish

Figure 6: F1-scores of the projected models for the four languages

Table 11 shows the results for the test sets. Unlike in the Finnish experiments, where the best model was picked by the performance on the development set, the model with the best performance on the test set is selected here. As seen in the Finnish tests, the results remain quite low for all of the languages. The weakest F1-score was achieved by Estonian, and as the Estonian data is missing the 'MISC'-category, and therefore has less possible categories to choose from than the other models, the results are even weaker by comparison. Spanish and Finnish get F1-scores of 54 and 57, respectively, and the best F1-score is achieved by the Dutch model (63).

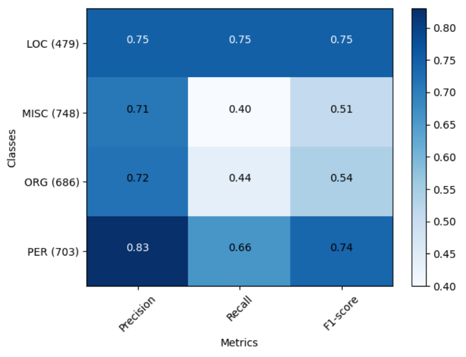
Table 11: Results for annotation projection models

	accuracy	precision	recall	F1-score
ES	92.5	59.4	49.8	54.2
ET	94.2	70.3	36.8	48.3
FI	96.5	59.1	56.3	57.7
NL	95.6	75.8	54.3	63.3

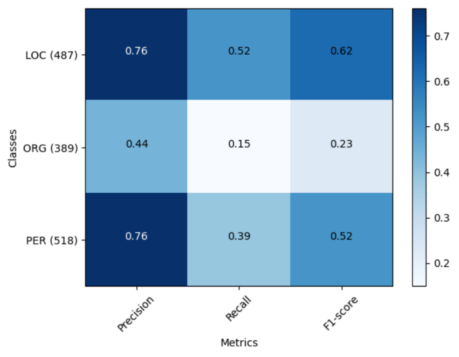
A more detailed depiction of the results, with results shown separately for each named entity type, is shown in Figure 7. For all the languages, location and person names were the type of named entities where the models performed better. The miscellaneous category was the most difficult for the models where it was included. For the Finnish test set, the 'MISC' category got an F1-score of 0.00, indicating that no instance of the miscellaneous category was classified correctly.

As mentioned in the data section, the Finnish corpus contained no 'MISC' tags originally, and I converted the product and event tags to 'MISC'. However, these were not common in the test set, and there was only 24 tokens classified as 'MISC' in the converted set. Classifying products and events as miscellaneous does correspond to the CoNLL annotation scheme, but the CoNLL 'MISC' category contains other kinds of tokens as well, such as adjectives such as 'Italian'. For these reasons, the model was not able to perform in this category. The 'MISC' category was difficult for the Spanish model as well, and it only achieved and F1-score of 10.

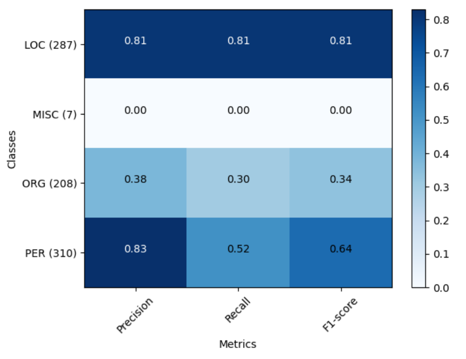
Interestingly, the models also performed weakly on the 'ORG' category. As seen in Table 8, the organization names are the most common type of named entity in the Finnish data, and are very likely to be the most common in the other languages as well. A lot of the projected data is EU data, and there organization names are very frequent. It might be that the organizations that occur in the test sets are simply very different than in the training data.



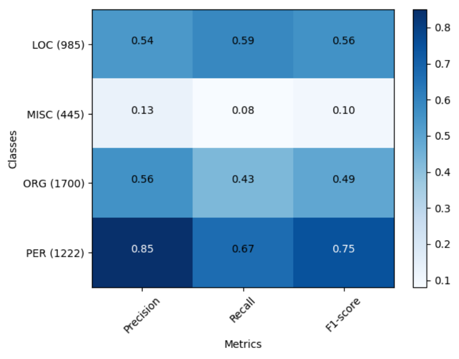
(a) Dutch



(b) Estonian



(c) Finnish



(d) Spanish

Figure 7: Detailed results for all four languages

The confusion matrices for all four languages can be seen in Appendix A, in Figures 12, 13, 14, and 15.

Experiments with annotated development set

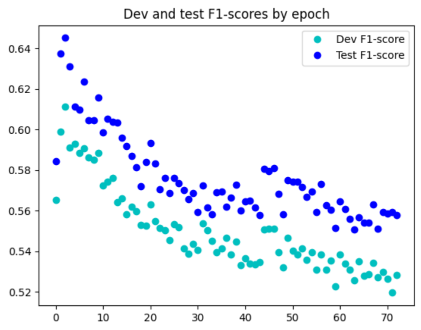
In the previous experiments conducted on Finnish projected data to find the best projection approach, there was a gradual decrease in the results along epochs. This was especially visible on the test set, but also occurred on the development set for most models. I thought this might be due to the fact that the training and development set were so different from the annotated test set. As I started building the models on projected data for all four languages, I also tried to use projected data only as the training set, and use annotated data as both the development and test sets. I thought this might improve the model's performance on the test set.

Results of this experiment are presented in Table 12. The models were trained for 48 hours. The projected data was created with the grow-diag with fixes approach here as well.

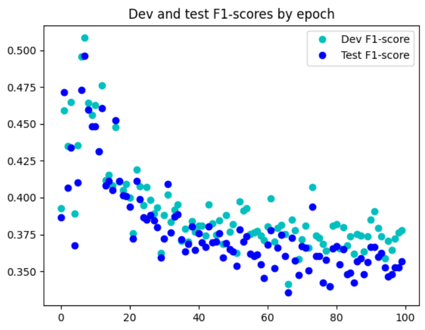
Table 12: Results for models with projected training data and annotated development set

	accuracy	precision	recall	F1-score	difference to main results F1-score
ES	92.2	60.5	45.4	51.9	-2.26
ET	94.3	70.1	38.4	49.6	+1.3
FI	96.6	57.9	56.9	57.4	-0.28
NL	95.8	74.0	57.2	64.5	+1.27

The results are quite similar for the test sets. For Spanish, using an annotated development set decreased the F1-score for about 2 percentage points, which was caused by a clear drop in the recall (from 49.8 to 45.4). For Estonian and Dutch, there was an improvement of about one percentage point, which is quite small and may be just random variation. For Finnish, the F1-score was virtually the same in both experiments. Figure 8 shows plots of the performance of the models over epochs for Dutch and Estonian. Here, the development set shows a similar decrease in results as the test set. This seems to suggest that the decrease in the results, already observed in the test sets in the previous experiment, is due to the annotated data being so different from the projected data. Results of Finnish and Spanish show a similar trend as well.



(a) Dutch



(b) Estonian

Figure 8: F1-scores of the projected models with annotated development sets

Analysis of annotation projection results

The performance of the models built on projected data was relatively poor, both compared to the state-of-the-art for the test sets used, and compared to previous research using similar methods. It would have been helpful to study a previous detailed account where annotation projection using a parallel word aligned corpus was conducted, but unfortunately I was not able to find one. It turned out that the basic method of just converting the tags from one language to another based on the word alignment mappings did not work so well. This was due to first of all the quality of the word alignments and sentence alignments in the corpus, which was not very high and contained many errors.

Additionally, more mistakes were introduced in several stages of the projection process. It seems that more complex heuristics would have been needed to deal with issues such as changing word orders within the named entities, and one-to-many and many-to-one links in the word alignments. Testing this kinds of fixes and finding ways to minimize errors caused by incorrect or incomplete annotations or word alignments proved tricky and time-consuming, especially as it had to be conducted for four languages.

Based on this experiment, I would say annotation projection would be best suitable for a restricted setting, where for example only one language pair is used. In that case, the projection can be more easily fine-tuned to take into account the issues specific to that language pair. On the other hand, the annotation projection process would benefit from an additional step where the correctness of the links between the words would be verified. This kind of a more thorough method for weeding out the mistakes would work better and would be easier to conduct, instead of writing several rule-based fixes to correct the mistakes.

Another bigger issue which decreased the results was that the training data and the test data were from different domains. The training and development data consisted of several genres, but mainly of EU press releases and medical documents, whereas the test sets consisted of news data. For these reasons, the results remained clearly below the state-of-the-art for all the languages.

Finally, in some cases projecting named entities from one language to another may result in flawed annotations, even though the system would function correctly. This is due to different traditions between languages in what is considered to be a named entity. For example, in the English CoNLL data set, adjectives denoting nationalities such as 'Spanish' are annotated as named entities of miscellaneous type. In the Finnish data set, such adjectives are not annotated as names, and considering them to be names in Finnish seems slightly counter-intuitive. Such differences may arise both from language-specific traditions, which affect especially more vague named entity categories such as miscellaneous names or events, and simply differences in annotation guidelines between human-annotated corpora. These issues are likely to decrease the results, and they may be partially responsible for the especially low results for the miscellaneous category.

5.2 Annotated BERT results

As I described earlier in the methods section, the second method I tried was training a multilingual BERT model on annotated data. I created a training and a development set by randomly mixing together all named entity annotated training and development data I had, in Spanish, Dutch, Estonian and Finnish. This resulted in a training set of 48,404 sentences and development set of 9,475 sentences. The BERT model did not have access to any information to which sentence was in which language.

The fine-tuning was conducted on my laptop on a CPU. To fine-tune the BERT model, I used the default settings used in the run_ner script. The training took 25 hours on a CPU, and the model was trained for three epochs. As I mixed the data sets together, I had to take into account the fact that the annotation scheme was slightly different in the data sets. For the Finnish data, the named entity tags marked as events were converted to MISC tags. For Estonian, there were no 'MISC' tags at all. This could not be corrected, so the annotations were not exactly the same in all the data.

Results for the model that had all four target languages in training, for each of the four test sets, are presented in Table 13. For the development set, the model got an accuracy of 98.8, recall was 88.3, precision was 88.0 and the F1-score was 88.2.

Table 13: Results for test sets for BERT models fine-tuned with annotated data

	acc	rec	prec	F1	F1 (Moon et al., 2019)	F1 (Luoma et al., 2020)
ES	97.8	85.8	86.4	86.1	**87.9**	-
NL	99.0	90.5	90.6	90.5	**91.1**	-
FI	98.9	86.8	85.4	86.1	-	**91.65**
ET	98.8	91.0	89.6	90.3	-	-

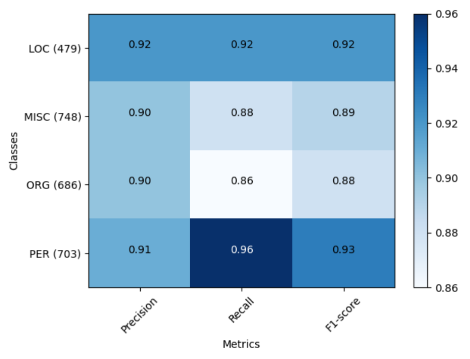
My hypothesis was that the F1-scores for Dutch and Spanish would be lower than in the reference article (Moon et al., 2019) due to introducing Finno-Ugric languages in the training set which are linguistically less related, but this was not the case. There was merely a two percentage point decrease in the F1-score for Spanish, and less than one percentage point decrease in the Dutch. For Finnish, the table contains the F1-score achieved by Luoma et al. (2020), which is a result of a mono-lingual BERT model, FinBERT[12]. The results are not directly comparable, as in that paper two more classes of named entities were included: products and dates. These are not annotated as named entities in the CoNLL annotation scheme, so they were left out.

My model achieved a slightly lower F1-score, perhaps due to a lack of an exhaustive grid search that Luoma et al. (2020) used to select some hyper-parameters. The pre-trained FinBERT also contains a customized WordPiece vocabulary, which covers much more Finnish words than multi-lingual BERT (Virtanen et al., 2019). This might affect the results, as Finnish is morphologically a fairly complex language. The lower performance of multi-lingual BERT compared to FinBERT is also in line with the results presented in Virtanen et al. (2019), where FinBERT was found to perform clearly better than multi-lingual BERT in most of the experiments tested.

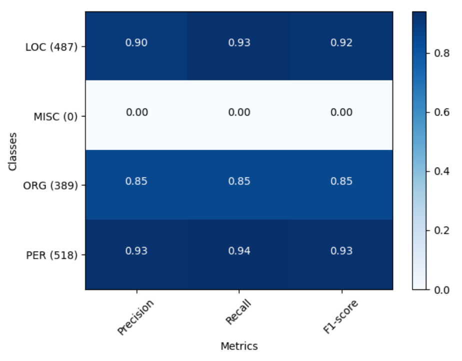
It should be noted that the Estonian scores are not directly comparable to the others, as it only contains 7 named entity categories instead of 9, and classification is therefore easier for the model.

Figure 9 shows performance per language per named entity type.

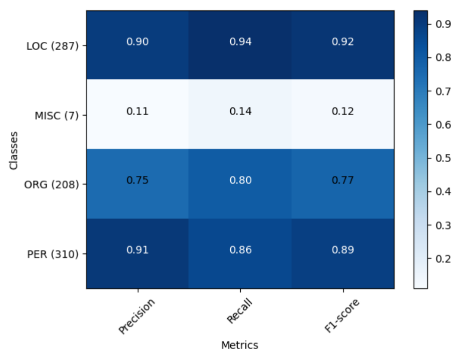
[12]http://turkunlp.org/FinBERT/



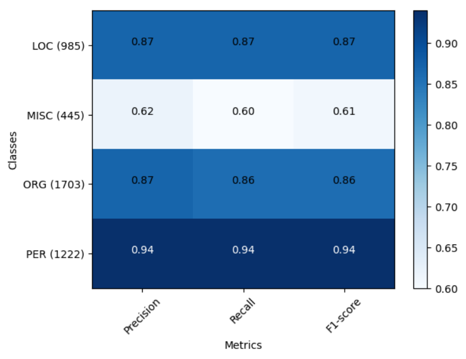
(a) Dutch



(b) Estonian



(c) Finnish



(d) Spanish

Figure 9: Detailed results BERT models for all four languages

Similarly to the annotation projection models, here the person and location names were the categories where highest F1-scores were achieved for all languages. For Dutch, the model performed steadily well in all categories. For Estonian, the ’MISC’ class received an F1-score of 0.00. This is due to the fact that no true ’MISC’ labels occur in the Estonian data. However, it did occur in the other languages’ training data, and the model predicted 7 tokens in the Estonian test data as ’MISC’. The discrepancy in the training data labels between languages did therefore cause some mistakes in the evaluation, but not many.

The miscellaneous category got the lowest results also for Finnish and Spanish. For Finnish, the number of instances in the test set is small, as already mentioned in the annotation projection results section. Additionally, Luoma et al. (2020) note that they also got clearly worse results for the event category, which here is included in the ’MISC’ category. They state that the test set contains difficult cases for this category.

5.3 Projected BERT results

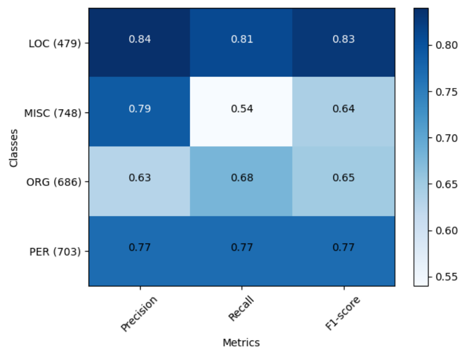
Next, I proceeded to fine-tune BERT with the projected data. I used 15k sentences per language for the training set, and 2,5k sentences per language for the development set. The results for the test set are visible in Table 14. For the development set, the accuracy was 96.9, precision was

66.1, recall was 63.8 and F1-score was 64.9.

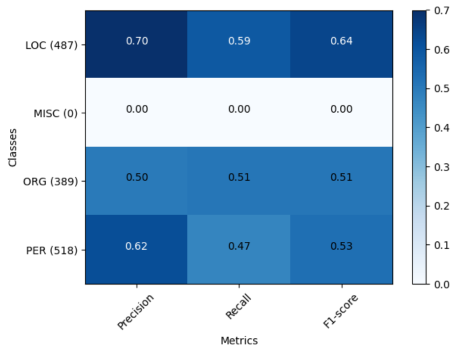
Table 14: Results for test sets for BERT models fine-tuned with projected data

	accuracy	recall	precision	F1-score
ES	92.6	55.8	51.7	53.7
NL	96.6	68.6	74.4	71.4
FI	96.2	64.5	48.5	55.4
ET	95.1	52.4	60.7	56.2

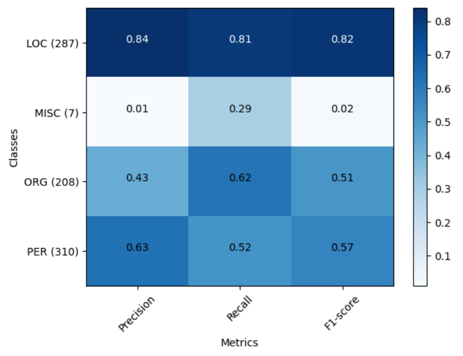
Figure 10 shows performance per language per named entity type.



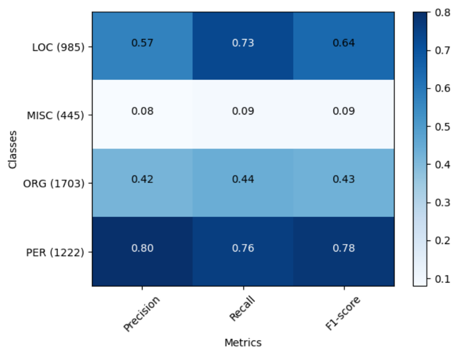
(a) Dutch



(b) Estonian



(c) Finnish



(d) Spanish

Figure 10: Detailed results for projected BERT models for all four languages

The results are clearly worse compared to the BERT models fine-tuned with annotated data. However, compared to the bi-LSTM models built with projected data, there are some clear improvements. For Dutch, the overall F1-score improves, and especially the recall is better for all categories. Also the two weakest-performing categories, 'MISC' and 'ORG', which got F1-scores of 51 and 54 in the bi-LSTM models, perform better here (F1-scores of 64 and 65). For Estonian, the biggest improvement was in the 'ORG'-category, where F1-score was 23 and here is 51.

For Finnish, 'PER' category got worse results here, but 'ORG' performed better, and even some 'MISC' entities were correctly labeled. For Spanish, there were no big differences in the performance of the different categories here compared to the projected bi-LSTM model.

5.4 Troubleshooting for annotation projection models: additional experiment

As both of the models using projected annotations performed poorly, I decided to make an additional experiment to shed light on the reasons for the low results. As stated in the section presenting the annotation projection results, the possible major reasons for the poor performance are the following:

1. Domain adaptation issues between training and test data
2. Mistakes in the source annotations
3. Annotation projection system problems

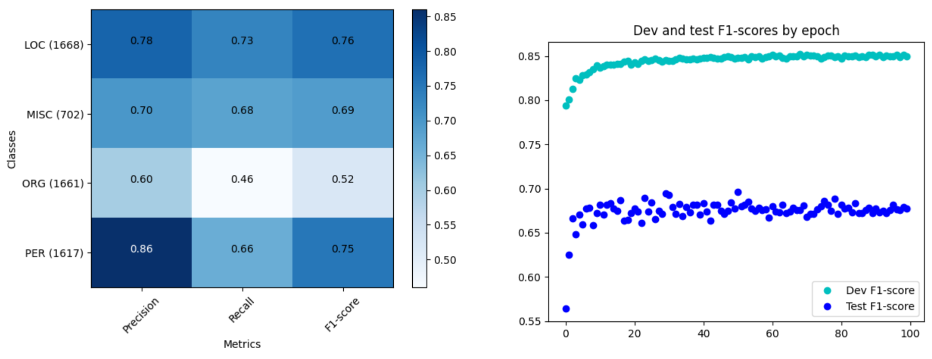
In order to find out which of the reasons affected the results the most, I trained another bi-LSTM model on the English TildeMODEL source data, annotated with the bert-base-NER model. In the methods section, I manually evaluated the quality of a small sample of the BERT model's annotations. Using that data as the training and development data, and testing it on the CoNLL English test set, I intended to find out whether the problems lie before or after the projection step. If this model would perform well, it would indicate that the domain adaptation issues and mistakes in the annotations do not affect the performance too much, and the problem is mistakes introduced in the annotation projection.

I trained a bi-directional LSTM model, using the same NCRF++ framework used in the main experiments. The training set consisted of 274,845 sentences, and the development set had 30,539 sentences. Both sets were bert-base-NER annotated English data, taken from the English-Finnish parallel corpus data extracted from the TildeMODEL corpus. The model was trained for 100 epochs, and tested on the CoNLL English test set.

Table 15 shows the results for the best model, selected based on the performance on the development set. Figure 11 shows the results for the model by named entity type, as well as the F1-scores of both the development set and the test set over epochs.

Table 15: Results for model trained with EN source data annotated with BERT

	accuracy	precision	recall	F1-score
dev set	98.8	86.0	84.4	85.2
test set	92.9	71.0	62.4	66.4



(a) Detailed results

(b) Performance over epochs

Figure 11: Detailed results for and performance over epochs for the English BERT annotations model

The results for the test set are clearly worse than what was achieved with the same NCRF++ framework by Yang and Zhang (2018). They used CoNLL training data as well as the CoNLL test set, and got an F1-score of 91.35 using the same parameters. Here, the best-performing model on the development set got an F1-score of 66.4 for the test set. This shows that the domain mismatch between the training and test data, together with the mistakes in the bert-base-NER annotations, does have a big impact on the low results for the projected models.

However, the results are slightly better than for the models built with annotation projection data. This model achieves an F1-score of 66.4 on the test set, whereas the only annotation projection model which came even close to that was the Dutch one, with an F1-score of 63.3. There's an even bigger difference in the results for the development sets. The projected models got F1-scores of roughly between 50 and 70, whereas here the F1-score for the development set is 85.2. This shows clearly that the annotation projection models fail to generalize well due to the noisiness of the data.

Interestingly, the plot of the performance over epochs in Figure 11 doesn't show a decrease in results over epochs for neither the test nor development set. This suggests that the overfitting in the projected models did not result from the domain mismatch between the test and training sets, but instead from the mistakes introduced in the projection phase. Perhaps the noisiness of the data played a part here as well, and as the model could not generalize well, it instead picked up some irrelevant tendencies from the data as training progressed.

5.5 Summary of results

Table 16 shows the F1-scores of all three approaches.

Table 16: A summary of the F1-scores of the three approaches

	LSTM with projected data	BERT with annotated data	BERT with projected data
ES	54.2	**86.1**	53.7
NL	63.3	**90.5**	71.4
FI	57.7	**86.1**	55.4
ET	48.3	**90.3**	56.2

The annotated multi-lingual BERT performed clearly the best. The two approaches, where projected data was used, yielded results clearly below the state-of-the art for all languages. When trained with projected data, multi-lingual BERT did not perform well due to the challenges in the quality of the projected annotations. However, compared to the bi-LSTM built on projected data, BERT with projected data performed better on the Dutch and Estonian test sets. For both languages, the F1-score was about eight percentage points higher for the projected BERT model than for the projected bi-LSTM. On the other hand, for Spanish and Finnish, the projected BERT performed slightly worse, with a drop in F1-score of about half a percentage point for Spanish, and more than two percentage points for Finnish.

Concerning the use of these methods in low-resource settings, it should be noted that even though the BERT method did not perform clearly better, it was able to achieve similar or better results than the bi-LSTM using much less training data per language, namely 15k sentences compared to 200-300 thousand.

6 Conclusions

In this thesis, I set out to test three methods of multi-lingual named entity recognition: building models using data created via annotation projection, training a multi-lingual BERT model and training multi-lingual BERT with projected annotations. First, I tested different methods to do annotation projection, and selected the best one. Then I built models, and tested their performance. The results were analyzed and compared them to each other, and reasons for the differences in the performance of the different models were considered. Finally, I did an additional experiment to study the performance of the projected annotations models.

The annotation projection proved to be relatively complicated, and more complex heuristics would have been needed to produce good results. Multilingual BERT performed well as expected, when fine-tuned with annotated data. Also, using the pre-trained BERT model, it proved to be a relatively painless process to build a well-performing multi-lingual NER model.

Additionally, it turned out that mixing together different language families, such as Germanic, Romance and Finno-Ugric, in the multilingual BERT's training data did not have a decreasing effect on the results when compared to previous approaches.

Further research

Based on the tests conducted in the thesis, it seems that multilingual BERT is the method that would prove the most fruitful to develop further. Using projected data to fine-tune the BERT model could prove to be feasible as well for languages where no training data is available. This would require developing a more reliable projection method than used here. Combining annotated data and projected data in fine-tuning the BERT model would also be interesting to test.

It would be interesting to apply multi-lingual BERT to languages with very few or no resources, to test its performance. More tests could be conducted to see how well multilingual BERT can deal with noisy data, or whether it could perform well with only a few hundred or a few thousand fine-tuning samples.

It would also be interesting to increase the number of languages in multilingual BERT's training data, and see how massively multi-lingual models can be built, and whether the performance would start decreasing at some point.

Appendices

A Confusion matrices of projected annotations models

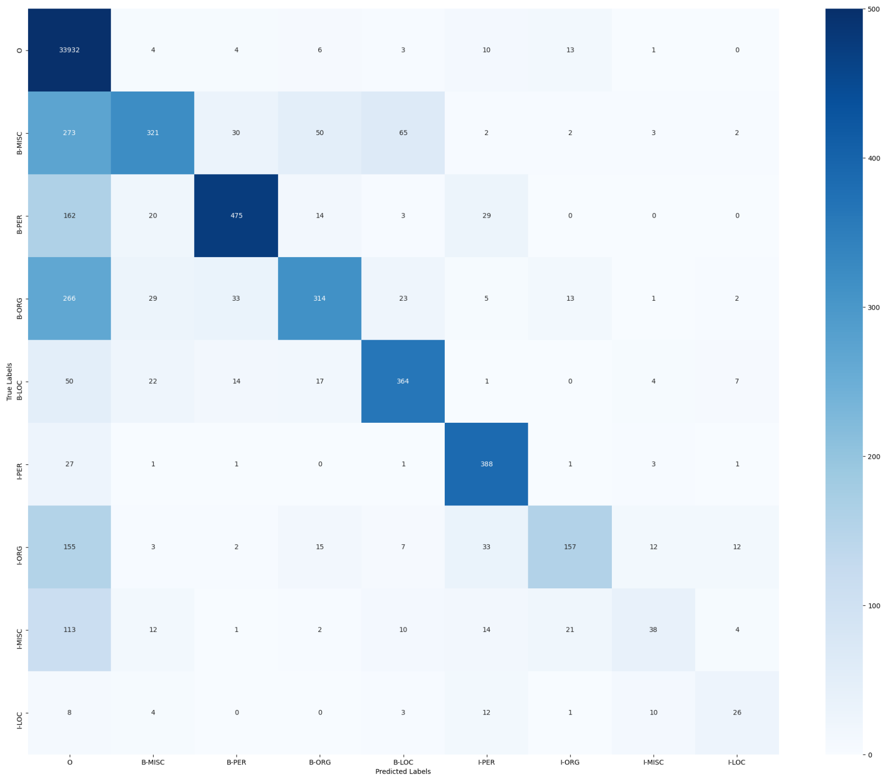


Figure 12: Confusion matrix for Dutch

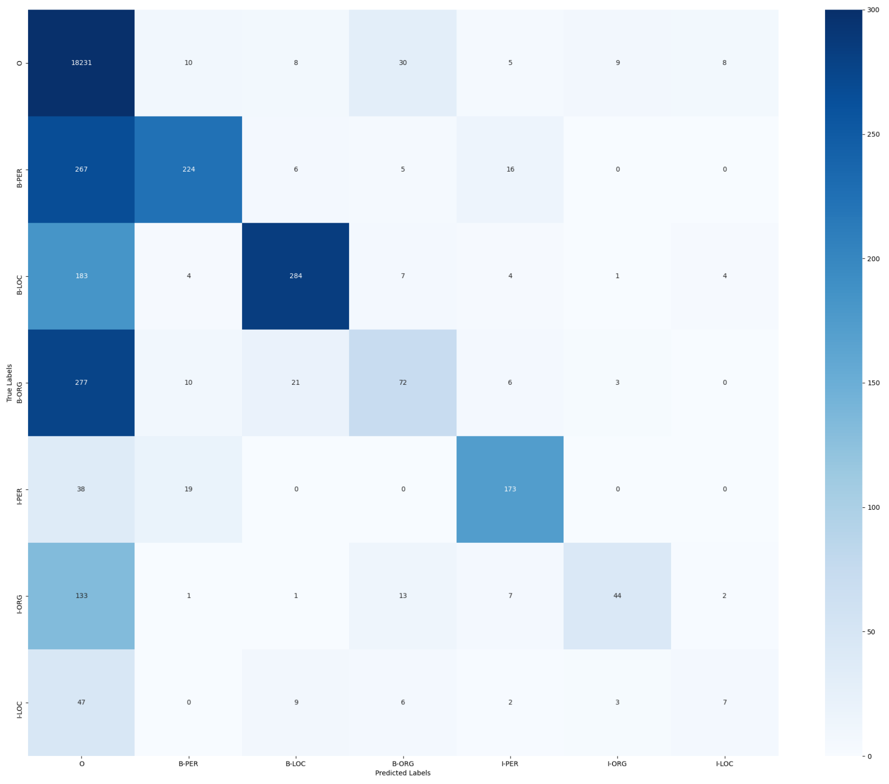


Figure 13: Estonian

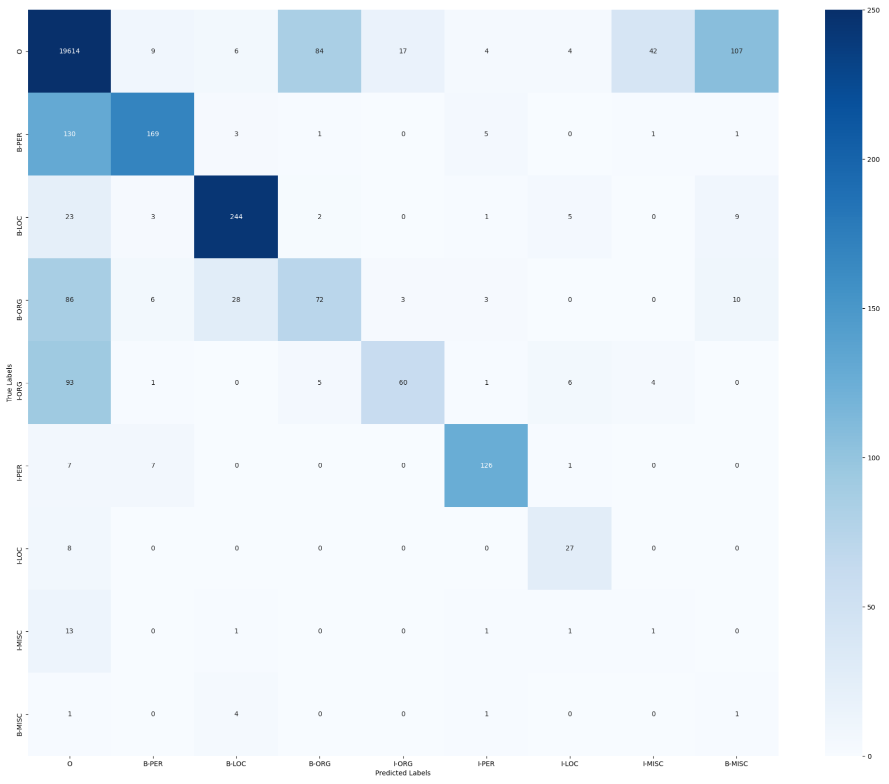


Figure 14: Finnish

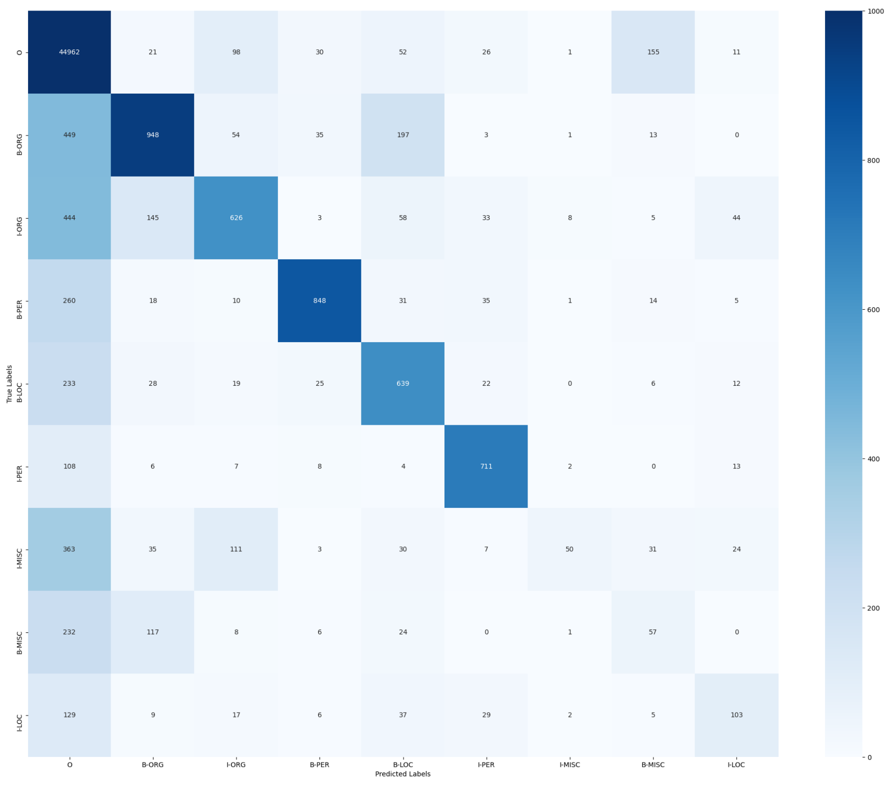
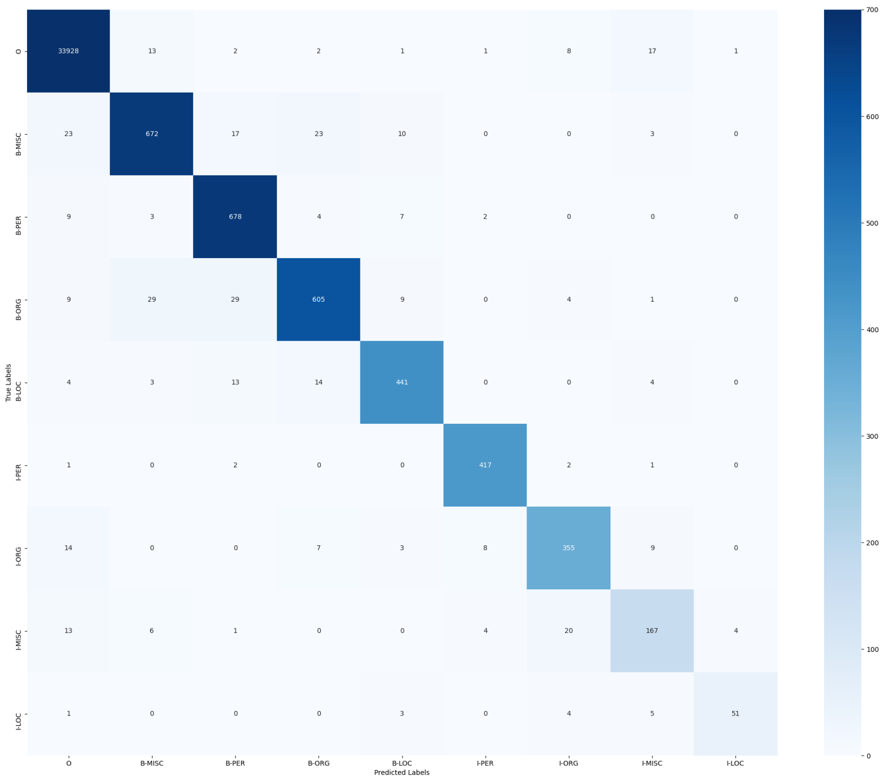


Figure 15: Spanish

B Confusion matrices of annotated BERT models



True Labels \ Predicted Labels	O	B-MISC	B-PER	B-ORG	B-LOC	I-PER	I-ORG	I-MISC	I-LOC
O	33928	13	2	2	1	1	8	17	1
B-MISC	23	672	17	23	10	0	0	3	0
B-PER	9	3	678	4	7	2	0	0	0
B-ORG	9	29	29	605	9	0	4	1	0
B-LOC	4	3	13	14	441	0	0	4	0
I-PER	1	0	2	0	0	417	2	1	0
I-ORG	14	0	0	7	3	8	355	9	0
I-MISC	13	6	1	0	0	4	20	167	4
I-LOC	1	0	0	0	3	0	4	5	51

Figure 16: Confusion matrix for Dutch

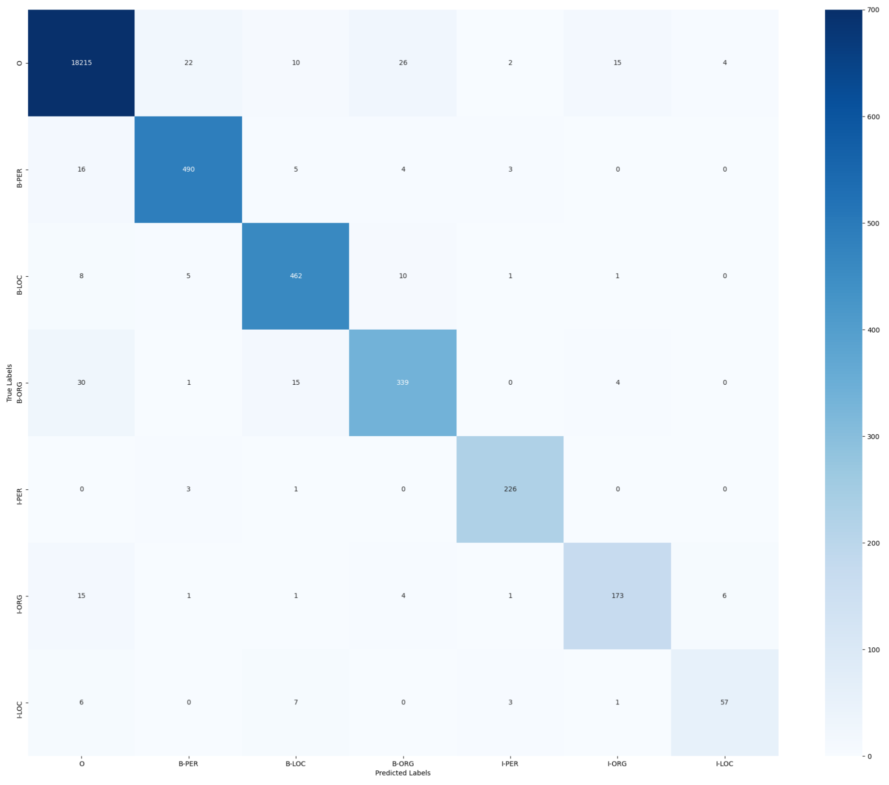


Figure 17: Estonian

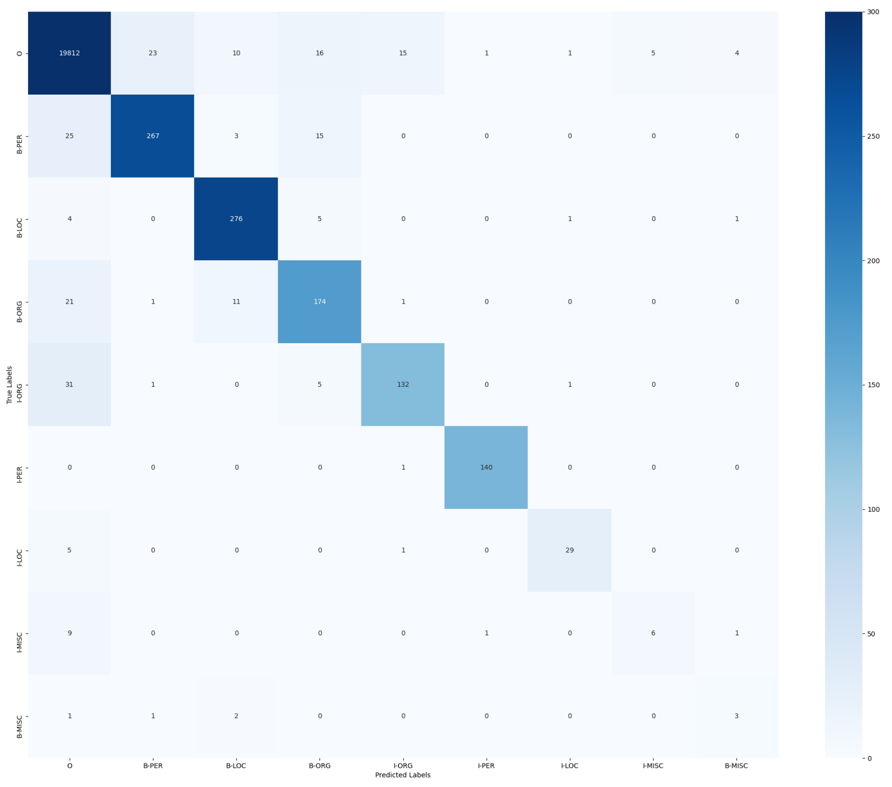


Figure 18: Finnish

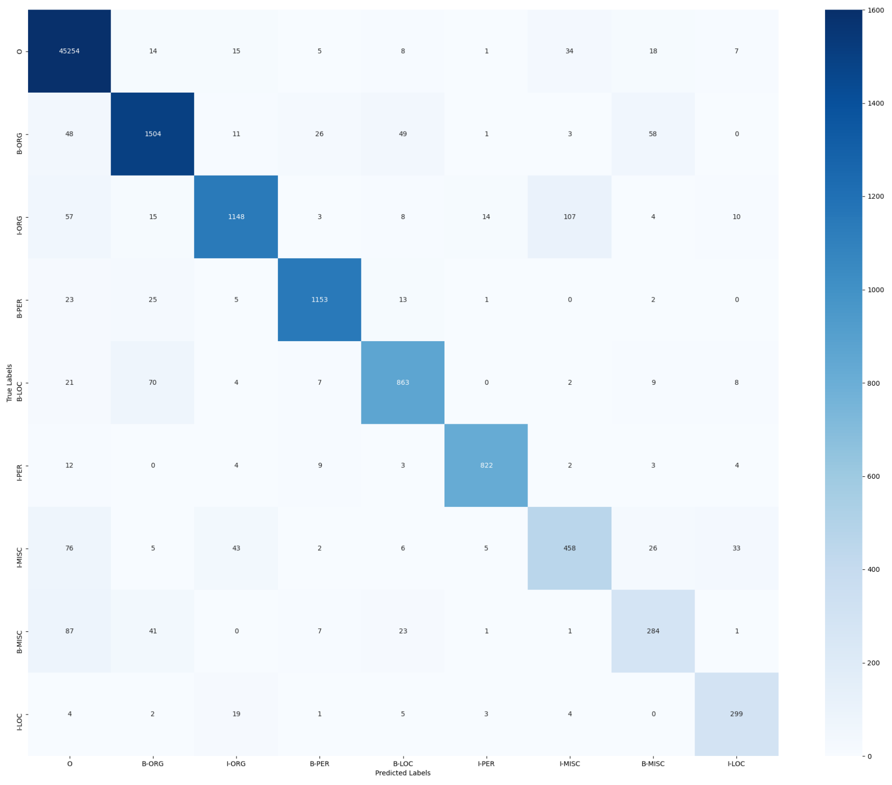


Figure 19: Spanish

C Confusion matrices of projected BERT models

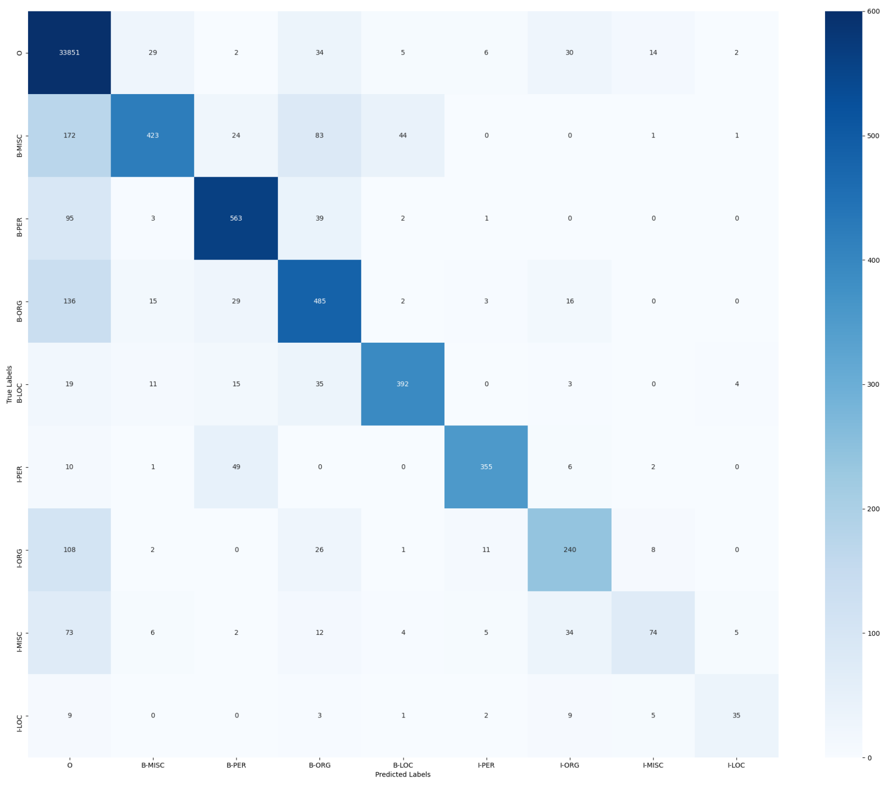


Figure 20: Confusion matrix for Dutch

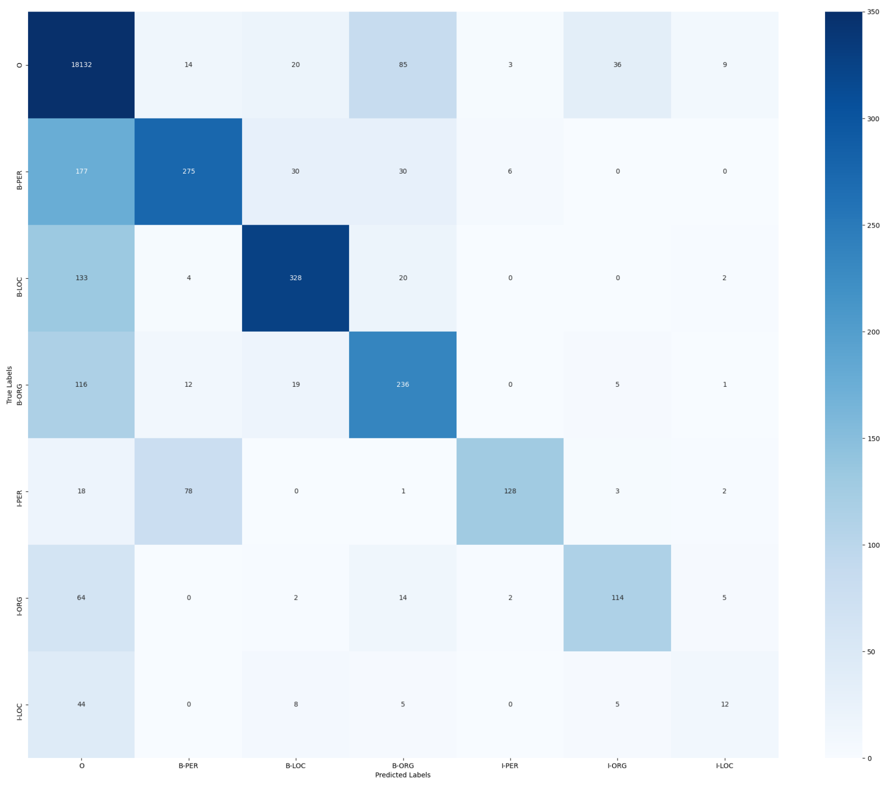


Figure 21: Estonian

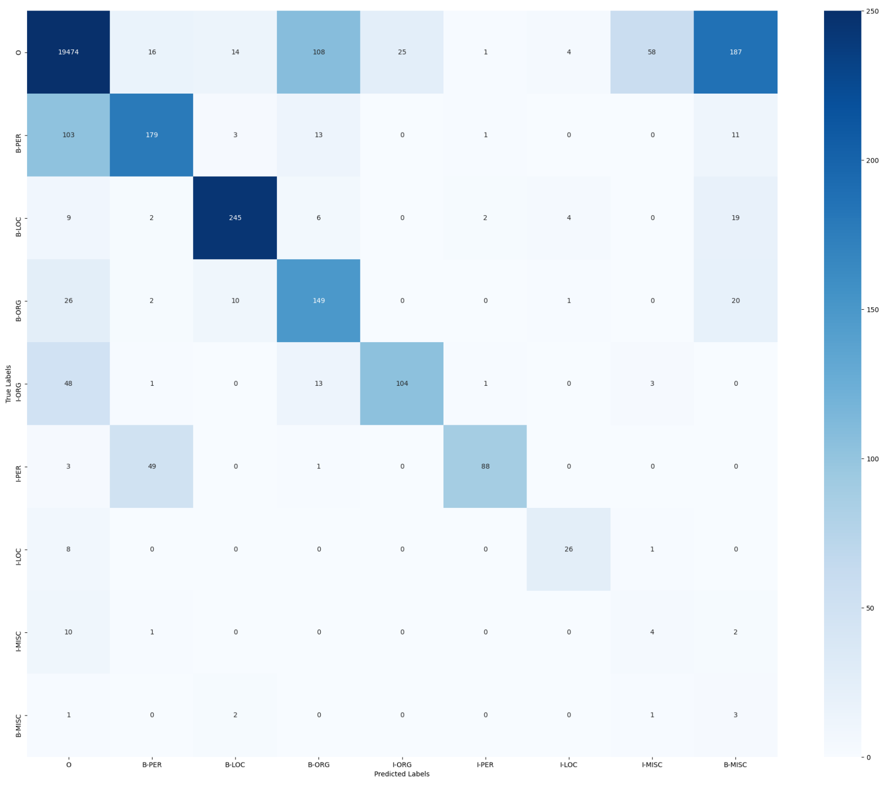


Figure 22: Finnish

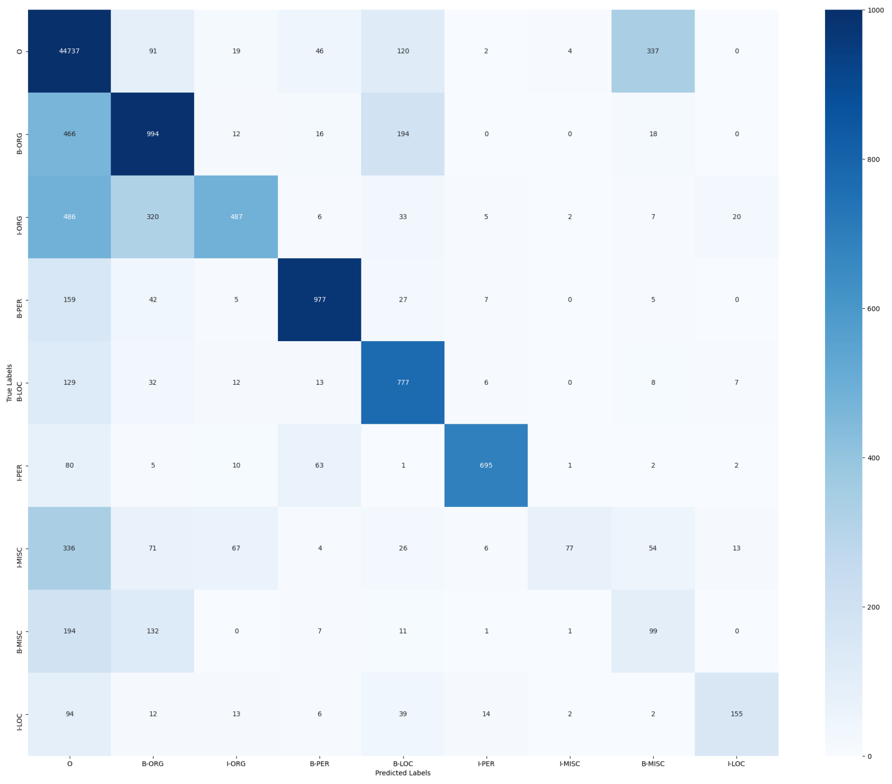


Figure 23: Spanish

References

Mikko Aulamo, Umut Sulubacak, Sami Virpioja, and Jörg Tiedemann. OpusTools and parallel corpus diagnostics. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 3782–3789. European Language Resources Association, May 2020. ISBN 979-10-95546-34-4. URL `https://www.aclweb.org/anthology/2020.lrec-1.467`.

Xilun Chen, Ahmed Hassan Awadallah, Hany Hassan, Wei Wang, and Claire Cardie. Multi-source cross-lingual model transfer: Learning what to share. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3098–3112, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1299. URL `https://www.aclweb.org/anthology/P19-1299`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. URL `http://arxiv.org/abs/1810.04805`.

Maud Ehrmann, Marco Turchi, and Ralf Steinberger. Building a multilingual named entity-annotated corpus using annotation projection. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2011*, pages 118–124, Hissar, Bulgaria, September 2011. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/R11-1017`.

Edouard Grave, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov. Learning word vectors for 157 languages. In *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

Ralph Grishman and Beth Sundheim. Message Understanding Conference- 6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996. URL `https://www.aclweb.org/anthology/C96-1079`.

Jue Hou, Maximilian Koppatz, José María Hoya Quecedo, and Roman Yangarber. Projecting named entity recognizers without annotated or parallel corpora. In *Proceedings of the 22nd Nordic Conference on Computational Linguistics*, pages 232–241, Turku, Finland, September–October 2019. Linköping University Electronic Press. URL `https://www.aclweb.org/anthology/W19-6124`.

Alankar Jain, Bhargavi Paranjape, and Zachary C. Lipton. Entity projection via machine translation for cross-lingual ner, 2019.

Jenna Kanerva, Juhani Luotolahti, V. Laippala, and F. Ginter. Syntactic n-gram collection from a large-scale corpus of internet finnish. In *Baltic HLT*, 2014.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. *CoRR*, abs/1603.01360, 2016. URL `http://arxiv.org/abs/1603.01360`.

Jouni Luoma, Miika Oinonen, Maria Pyykönen, Veronika Laippala, and Sampo Pyysalo. A broad-coverage corpus for Finnish named entity recognition. In *Proceedings of the 12th Language Resources and Evaluation Conference*, pages 4615–4624, Marseille, France, May 2020. European Language Resources Association. ISBN 979-10-95546-34-4. URL `https://www.aclweb.org/anthology/2020.lrec-1.567`.

Xuezhe Ma and Eduard H. Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf. *CoRR*, abs/1603.01354, 2016. URL `http://arxiv.org/abs/1603.01354`.

Taesun Moon, Parul Awasthy, Jian Ni, and Radu Florian. Towards lingua franca named entity recognition with bert, 2019.

Rudra Murthy, Mitesh M. Khapra, and Pushpak Bhattacharyya. Improving ner tagging performance in low-resource languages via multilingual learning. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 18(2), December 2018. ISSN 2375-4699. doi: 10.1145/3238797. URL `https://doi.org/10.1145/3238797`.

Hiroki Nakayama. seqeval: A python framework for sequence labeling evaluation, 2018. URL `https://github.com/chakki-works/seqeval`. Software available from https://github.com/chakki-works/seqeval.

Malarkodi Nathan and S. DEVI. Generic feature selection methodology to named entity detection from indian and european languages. *Advances in Electrical and Computer Engineering*, 19:79–88, 02 2019. doi: 10.4316/AECE.2019.01011.

Telmo Pires, Eva Schlinger, and Dan Garrette. How multilingual is multilingual BERT? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4996–5001, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1493. URL `https://www.aclweb.org/anthology/P19-1493`.

Afshin Rahimi, Yuan Li, and Trevor Cohn. Massively multilingual transfer for NER. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 151–164, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1015. URL `https://www.aclweb.org/anthology/P19-1015`.

Shruti Rijhwani, Shuyan Zhou, Graham Neubig, and Jaime Carbonell. Soft gazetteers for low-resource named entity recognition. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8118–8123, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.722. URL `https://www.aclweb.org/anthology/2020.acl-main.722`.

Roberts Rozis and Raivis Skadiņš. Tilde MODEL - multilingual open data for EU languages. In *Proceedings of the 21st Nordic Conference on Computational Linguistics*, pages 263–265, Gothenburg, Sweden, May 2017. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W17-0235`.

Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. A finnish news corpus for named entity recognition. *Language Resources and Evaluation*, 54(1):247–272, Aug 2019. ISSN 1574-0218. doi: 10.1007/s10579-019-09471-7. URL `http://dx.doi.org/10.1007/s10579-019-09471-7`.

Jörg Tiedemann. Parallel data, tools and interfaces in opus. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Mehmet Ugur Dogan, Bente Maegaard, Joseph

Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey, may 2012. European Language Resources Association (ELRA). ISBN 978-2-9517408-7-7.

Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: Language-independent named entity recognition. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*, 2002. URL `https://www.aclweb.org/anthology/W02-2024`.

Erik F. Tjong Kim Sang and Fien De Meulder. 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, 2003. URL `https://www.aclweb.org/anthology/W03-0419`.

Alexander Tkachenko, Timo Petmanson, and Sven Laur. Named entity recognition in Estonian. In *Proceedings of the 4th Biennial International Workshop on Balto-Slavic Natural Language Processing*, pages 78–83, Sofia, Bulgaria, August 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/W13-2412`.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.

Antti Virtanen, Jenna Kanerva, Rami Ilo, Jouni Luoma, Juhani Luotolahti, Tapio Salakoski, Filip Ginter, and Sampo Pyysalo. Multilingual is not enough: BERT for finnish. *CoRR*, abs/1912.07076, 2019. URL `http://arxiv.org/abs/1912.07076`.

Ralph Weischedel, Eduard Hovy, Mitchell Marcus, Martha Palmer, Robert Belvin, Sameer Pradhan, Lance Ramshaw, and Nianwen Xue. *OntoNotes: A Large Training Corpus for Enhanced Processing*. 01 2011.

Qianhui Wu, Zijia Lin, Börje F. Karlsson, Biqing Huang, and Jian-Guang Lou. Unitrans : Unifying model transfer and data transfer for cross-lingual named entity recognition with unlabeled data. In Christian Bessiere, editor, *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI-20*, pages 3926–3932. International Joint Conferences on Artificial Intelligence Organization, 7 2020. doi: 10.24963/ijcai.2020/543. URL `https://doi.org/10.24963/ijcai.2020/543`. Main track.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, ukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, and Jeffrey Dean. Google's neural machine translation system: Bridging the gap between human and machine translation. 09 2016.

Jie Yang and Yue Zhang. NCRF++: An open-source neural sequence labeling toolkit. In *Proceedings of ACL 2018, System Demonstrations*, pages 74–79, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-4013. URL `https://www.aclweb.org/anthology/P18-4013`.