Projecting named entity recognizers without annotated or parallel corpora

Jue Hou, **Maximilian W. Koppatz**,
José María Hoya Quecedo and **Roman Yangarber**
University of Helsinki, Department of Computer Science, Finland
first.last@helsinki.fi

Abstract

Named entity recognition (NER) is a task extensively researched in the field of NLP. NER typically requires large annotated corpora for training usable models. This is a problem for languages which lack large annotated corpora, such as Finnish. We propose an approach to create a named entity recognizer for Finnish by leveraging pre-existing strong NER models for English, with no manually annotated data and no parallel corpora. We automatically gather a large amount of *chronologically matched* data in the two languages, then project named entity annotations from the English documents onto the Finnish ones, by resolving the matches with simple linguistic rules. We use this "artificially" annotated data to train a BiLSTM-CRF NER model for Finnish. Our results show that this method can produce annotated instances with high precision, and the resulting model achieves state-of-the-art performance.

1 Introduction

The goal of Named Entity Recognition (NER) is to recognize names and classify them into pre-defined categories, based on their context. The quality of NER is crucial, since it is an important step in modern NLP, e.g., information retrieval (IR) or information extraction (IE) systems. Various approaches have been proposed to tackle the NER task, including (Finkel et al., 2005; Huang et al., 2015; Lample et al., 2016; Ma and Hovy, 2016; Chiu and Nichols, 2016; Reimers and Gurevych, 2017; Peters et al., 2018; Devlin et al., 2018). These approaches require large annotated datasets to train models, and have been shown to be effective for languages with abundant linguistic resources, such as English.

However, not all languages are as resource-rich as English. There are significantly fewer resources for languages such as Finnish. Further, very few NER taggers or corpora are publicly available online. The FiNER tagger from the Language Bank of Finnish[1] is one of the few, but we found no documentation of its performance.

Automatically annotating corpora for training NER models is one solution to this problem. Several approaches have been proposed for building such corpora for NER. Most of these rely on the Wikipedia corpus, (Al-Rfou et al., 2015; Ghaddar and Langlais, 2017; Kim et al., 2012; Richman and Schone, 2008; Kazama and Torisawa, 2007; Toral and Munoz, 2006; Nothman et al., 2013). However, the amount of Wikipedia documents in Finnish is also relatively small.

In this paper, we propose a novel approach for automatically marking Finnish text with NE annotations, for the purpose of training a statistical NER model from these annotated data. This can be viewed as a *projection* of a pre-existing NER model in one language to a NER model in another language. The core idea of our annotation approach is to utilize strong NER available for English and to match automatically annotated English data with Finnish data by resolving the base form of names. Ehrmann et al. (2011) proposed an idea of model projection similar to the one in the this work. However, rather than resolving the base form of named entities in target language internally as we do, they used machine translation as the basis for projection. This allows them to project models between different languages, including in languages with different writing systems, such as Russian and English. However, this assumes the existence of a high-quality *machine translation* system, and token binding between the languages, which determine the quality of the NER training dataset.

Using the resulting annotated data, we train an BiLSTM-CRF model on the basis of (Ma and Hovy,

[1] www.kielipankki.fi

Proceedings of the 22nd Nordic Conference on Computational Linguistics (NoDaLiDa), pages 232–241
Turku, Finland, 30 September – 2 October, ©2019 Linköping University Electronic Press

2016; Reimers and Gurevych, 2017), and evaluate it on a *manually annotated* dataset. Our results show that training models on data annotated in this way achieves improved performance for Finnish NER tagging over models trained on the publicly available data alone. This suggests that our approach works well for annotating a corpus with named entities automatically, and enables using this corpus to learn good-quality NER models for less-resourced languages.

The paper is organized as follows. In section 2 we briefly introduce a few terms and key concepts used throughout the paper. In section 3 we describe the data sources, pre-processing steps and the rule-based annotation pipeline. Section 4 describes our model architecture, as well as the parameters used in training. In section 5 we discuss the results obtained from the experiments. Section 6 concludes with current directions of research.

2 Terminology

Base form: The base form of a word, also is referred as *lemma*, is the canonical, or "dictionary" form of a word. For example, the base form of the English token "was" is "be."

Surface form: The surface form of a token is the form in which the word appears in the actual text. Words in this form may be inflected, such as "was," or be identical with its base form, such as the past participle of "run".

Compound: A compound is a word which consists of multiple root morphemes. For example, "pancake" consists of two parts: "pan" and "cake". Some languages, such as Finnish and German, make extensive use of compounding.

3 Automatic projection pipeline

3.1 Data source

Finnish names and their types are obtained by matching the base forms of English names with Finnish *potential* names. This section details this procedure step by step.

English news gathering and name processing: English news is collected by our business news surveillance system, PULS (Du et al., 2016), from over 3,000 English sources.[2] Over 5,000 documents are gathered daily. Each document collected by the system is processed by a cascade of pre-processing classifiers, including a pattern-based named entity tagger. Here, we obtain the base forms of names and their types, which are later used for projection.

Source	prec	rec	F1
PULS pattern-based	0.68	0.37	0.30
BiLSTM-CRF-GloVe	0.87	0.85	0.85
BiLSTM-CRF-W2V	0.89	0.90	0.89

Table 1: English NER tagger quality on CoNLL2003 test dataset

The performance, especially the *precision*, of the English NER tagger is therefore crucial for the entire pipeline. It is worth pointing out that the precision of the English NER tagger controls the quality of the projected Finnish data. The recall, on the other hand, determines the variety of the projected named entities. Lower recall rate can be compensated by feeding in more news articles. Therefore, in this paper, the precision of English NER tagger is considered to be more essential than the recall or the overall F1 score.

For comparison, we also trained two BiLSTM-CRF models (Ma and Hovy, 2016) from scratch, using Word2Vec and GloVe word embeddings. These models were trained on the CoNLL2003 English dataset. Table 1 shows an evaluation of all three English NER taggers on the CoNLL2003 test dataset.

We should mention that the PULS NLP system has different tokenization compared to the CoNLL dataset, and our pattern-based NER tagger is customized for the *business* news domain. Though the output of our pattern-based tagger is aligned to be comparable with the CoNLL dataset, the content of its test dataset, which is mostly sports news, is still skewed against our tagger. In practice, our tagger achieves higher precision on business news. To confirm this, we evaluate the PULS tagger on 10 randomly selected articles, containing both general and business news. Although this is a simple experiment, the overall precision of the PULS tagger increases to 77%. As for the two BiLSTM-CRF models, the results are different from what was reported in the papers (Ma and Hovy, 2016; Reimers and Gurevych, 2017), since we used a default hyper-parameter setup, rather than using the fine-tuned setup in the papers.

Finnish news gathering and name pre-processing: Finnish news articles are collected

[2] http://newsweb.cs.helsinki.fi/

from two major Finnish online news agencies. Around 200 news articles are collected daily. The Turku dependency parser (Haverinen et al., 2014) has been applied for sentence splitting and tokenization. Three different problems need to be solved so that all potential names can be extracted in these steps: name identification, base form selection, and name merging.

Name identification: For identifying whether a token is a name or part of a name, we use the rules based on the position of tokens as follows:

- Any capitalized token which appears in the middle of a sentence is definitely a name or part of the name.
- If token A is a name according to previous rule, and token B, having the same base form as token A, appears at the beginning of a sentence, we assume token B is the same name.
- If a token of a potential name appears in the document only at the beginning of sentences, it is not certain and therefore not assumed to be a name.

Base form selection: To determine the base form corresponding to a surface form found in text, we consider all base forms returned by our morphological analyzer, (Moshagen et al., 2013), and a simple rule-based stemmer, and look through the entire article. If there is an intersection between the possible base forms of two name tokens, their true base form can then be resolved. When the intersection has only one base form, it can be confirmed to be the base form of a name directly. Otherwise, all of them will be recorded as potential base form of the name. All potential base forms will be further filtered when matching with English named entities during projection.

Suppose the surface form "Trumpille" (in the allative case) and "Trump" (nominative) both exist in an article. Without any external knowledge, the Finnish surface form "Trumpille" will be assigned two potential lemmas by our stemmer: "Trumpi" and "Trump" (both of these lemmas have the same allative form). For the surface form "Trump", only "Trump" will be returned as a potential lemma. Then, in this case, their intersection, "Trump", will be confirmed to be the lemma of both "Trumpille" and "Trump". However, if instead the article only contains the surface forms "Trumpille" and "Trumpin" (genitive). Then both lemmas "Trumpi" and "Trump" will be recorded as potential lemmas.

We perform name identification and base form selection jointly, since they are connected, by searching for the common base forms of tokens.

Name merging: We use a set of rules to merge names that consist of more than one token. Potential names can contain only the following kinds of tokens in positions other than the final position:

- Singular common noun or proper noun which must be in the nominative case, for example: "Spring Harbour".[3]
- English function words: e.g., "the", "and", "new", etc. For example: "The New York Times"
- Having no valid analyses returned by the Finnish morphological analyzer, and its surface form can be confirmed to be its own lemma during the base form selection process above.
  One example is the token "Trump" in "Trump Towerin" (genitive: "of the Trump Tower"). Our Finnish morphological analyzer will reject (not recognize) the input token "Trump". We assume that "Trump" can be confirmed as a name or as part of a multi-token name according to the rules. "Trump" can be confirmed to be its own base form, which means its base form happens to be the same as its surface form. In this case, "Trump" will be merged with the following token "Towerin".

We should note that when several potential name tokens are strung together, the true partitioning of names is ambiguous. During name merging, all different partitionings and potential forms of the base forms of names are cached as candidates for the following name resolution step.

Hyphenating between tokens are also a criterion for merging names, such as the Indian surname "Ankalikar-Tikekar".

3.2 Name projection

In the next stage, we annotate Finnish names by utilizing the potential names candidates produced by the previous three steps, namely name identification, base form selection, and name merging.

[3] Names such as "Helsingin Sanomat" (name of a major newspaper in Finland), where the first token is in the *genitive* case (of "Helsinki") are currently not handled by these rules, and are handled separately by a list.

The fundamental assumption is that *a name refers to only one entity in a given article*. We expect this assumption to hold for well-edited news articles. This means that if only one instance (surface form) of a particular name has been annotated in an article, the remaining occurrences of the same name in the article—possibly involving other surface forms—can be annotated as well.

We gather two sets of named entities from Finnish document and English documents:

- For a Finnish document, published on day t, we use three steps mentioned above to obtain a set of potential Finnish named entity candidates, including both potential base forms and confirmed base form of names.
- From English news in the time interval ($t \pm 2$ days), using an English NER tagger, a set of English named entities and their corresponding tags are obtained. Each of them has its base form resolved by the pre-processing pipeline in PULS.

Names can naturally be matched according to their base form. The type of the English named entities can therefore be projected to their Finnish counterparts. The remaining Finnish names candidates, for which no type annotation can be inferred, are dropped after this step.

The idea of a time window ($t \pm 2$ days) is to take advantage of the fact that names overlap significantly in different articles due to continuous coverage of important events, and therefore optimize our memory usage and time efficiency.

Again, take "Trump" as an example. Suppose we have a named entity "Donald Trump" from the English news articles and it is recognized as "Person". We may have "Donald Trumpille" in a Finnish article; if the surface form "Trump" is not present in the same Finnish article, as we mentioned already, we can only infer that the base form of "Trumpille" is "Trumpi" or "Trump", using stemming rules. In addition, "Donald Trumpille" has two tokens but we do not yet know whether they belong together as one name. Therefore, "Donald", "Donald Trump", "Donald Trumpi", "Trump" and "Trumpi" are all Finnish name *candidates*. After matching, only "Donald Trump" will be kept and annotated as *Person*, while other candidates, namely "Donald", "Donald Trumpi", "Trump" and "Trumpi", are dropped.

In addition, for the *Person* type only, names will be connected by their partial base form. Once "Donald Trump" gets annotated, all the other "Donald" and "Trump" tokens in the entire article will be annotated as *Person* as well.

3.3 Special cases: rule-based projection

We use extra steps to handle special cases in this process. In Finnish, geo-location names, such as the names of countries, are often different from their English names. For example, France is "Ranska" in Finnish, and the United States is "Yhdysvallat" in Finnish. Some organizations also have the same problem, as UN is "YK" in Finnish, etc. Therefore, we manually build a small database of frequent names, including Finnish geo-locations, and a few of the major and most frequently occurring international companies and organizations, to assure that they are annotated correctly. In addition, this covers some cases which the English tagger fails to catch. We also filter out names that can have multiple types, such as MacLaren, since these are ambiguous.

Additionally, we introduce a list of 1000 common first names and assume that names beginning with these tokens are of type *Person*. However, this practice requires more rules to constrain its outcome:

- A Person name should have at most 2 tokens.
- A Person name should not start with "The".
- No token in a Person name should be fully uppercase.
- We require that a Person name be mentioned using the full name at least once in the article.

These rules are simple, naive and strict. The purpose of these rules is to remove any uncertain instances and make the data as clean as possible. Even if only one name in an article can meet all these rules, all other name instances related to that name instance will be correctly annotated. Also, taking advantage of our enormous amount of data, we can afford to filter out uncertain data without worrying about the amount of remaining data.

Currently, the annotations may be wrong when an article only mentions the last name of a person, which also happens to be the name of a location. For example, "Sipilä" is the last name of the current Prime Minister of Finland, and may therefore be mentioned many times in an article, without mentioning the full name, “Juha Sipilä”. Coincidentally, “Sipilä” is a town in Finland. The situation where both the person and the location are mentioned in the same article rarely occurs in practice and can be tackled by filtering out such names.

4 NER model

Next we provide the details of the adapted BiLSTM-CRF model for Finnish NER and the hyperparameter setup for training this model. The basic network structure of the model is inspired by (Ma and Hovy, 2016; Reimers and Gurevych, 2017). The model is implemented in Keras with TensorFlow as its backend. The CRF layer is provided by Keras-contrib.[4] The training process was run on an Nvidia GeForce 1080 Ti GPU. It took around 3 hours to train the model using the setup in this section. The model is shown in Figure 1.

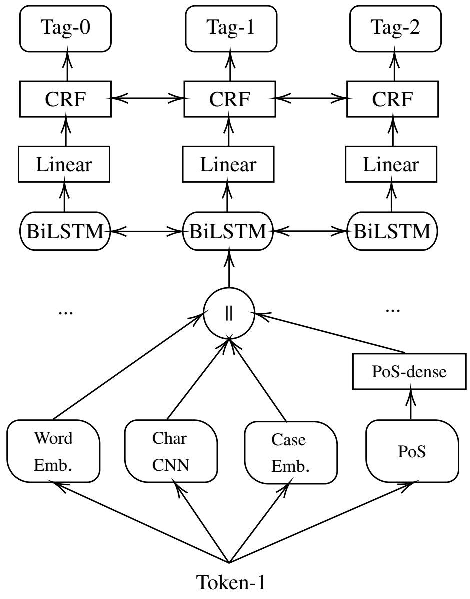

Figure 1: Adapted BiLSTM-CRF network structure for Finnish NER

As seen in Figure 1, Part-of-Speech (PoS) is included as an additional feature, compared to the model of Ma and Hovy (2016). This is because a lemma may be assigned multiple PoS tags by our morphological analyzer (Moshagen et al., 2013). Word embeddings such as Word2Vec (Mikolov et al., 2013) may implicitly contain PoS information but will still be static regardless of context. Using PoS as input feature also compensates for out-of-vocabulary problem in embeddings. In these cases, not even the implicit PoS information can be detected by the network if PoS is not a part of input features.

4.1 Data encoding

Tokens are encoded into to several features: *word embedding*, *character embedding*, *case embedding* and *Part of Speech* (PoS). Except for PoS, most of the features follow the setup in (Reimers and Gurevych, 2017). Word embeddings are extended with a special mark for ambiguous tokens—tokens, for which our morphological analyzer fails returns more than one base forms and PoS. These tokens are replaced with a special token “AMBIGUOUS”. Additionally, we only use the embedding of the last part of a compound word if this word is out-of-vocabulary. This is because the last part is the essential part of compounds in Finnish. Character embeddings are extended with a special value for “unrecognized” character. The PoS feature is encoded as an array of ones and zeros. Each dimension corresponds to one PoS type, including PADDING and UNKNOWN. Integer “1” is assigned to the dimension corresponding to the token’s PoS. If a token is a compound word, only the PoS of its last part is used for encoding. If a token has multiple PoS analyses, more than one position in the PoS array is assigned “1”. The values of PoS are as follows:

- PADDING
- UNKNOWN
- Noun
- Verb
- Adj
- Adv
- Pron
- Conj
- Interj
- Num
- Punct
- other

These four input features are concatenated before feeding into BiLSTM.

4.2 Parameter initialization

Word embeddings: We use a pre-trained Word2Vec embedding matrix, which is trained by (Laippala and Ginter, 2014). It has been trained on 4.5B words of text. As mentioned previously, we include vectors for “PADDING”, “UNKNOWN” and “AMBIGUOUS” tokens. Embeddings for the tokens “UNKNOWN” and “AMBIGUOUS” are randomly initialized with uniform sampling from -0.25 to 0.25, while the “PADDING” embedding is a zero vector.

[4] www.github.com/keras-team/keras-contrib

Layer	*Hyper-parameter*	*Number*
Char CNN	Number of filters	30
	Filter size	3
PoS-dense	Unit size	30
	Activation	Relu
BiLSTM	Number of layers	2
	State size	200
	Dropout rate	0.25

Table 2: Table of hyper-parameter for experiments

Character embeddings: Character embeddings, including "UNKNOWN" character embedding, are randomly initialized with uniform samples from $-\sqrt{\frac{3}{dim}}$ to $\sqrt{\frac{3}{dim}}$, where $dim = 30$.

Case embedding: Case embeddings are randomly initialized applying a uniform initializer. The dimensionality of the case embeddings is 10.

Weight Matrices and Bias Vectors: Most of weights are initialized as a uniform sample from $[-\sqrt{\frac{6}{N_i+N_o}}, \sqrt{\frac{6}{N_i+N_o}}]$, where N_i and N_o refer to the number of input and output units in the weight tensor. Bias is initialized with zeros.

4.3 Optimization

Optimizer: We used the Adam optimizer, as recommended in (Reimers and Gurevych, 2017). The setup for the Adam optimizer also followed the Keras default setting: $lr = 0.001, \beta_1 = 0.9, \beta_2 = 0.999$. Although Ma and Hovy (2016) used gradient norm clipping of 5.0, we did not.

Early stopping and learning rate decay: We applied early stopping following the categorical accuracy on the training dataset in case of overfitting. On average, the training process stops after 5 epochs. We have also explored reducing the learning rate during the training process if the accuracy stops improving. However, this made the training slower, and did not improve the final result on the validation dataset.

4.4 Hyper-parameter setup

Most of the hyper-parameter values, shown in Table 2, follow the recommendations in (Reimers and Gurevych, 2017). The layer called "PoS-dense" in Figure 1 is a dense layer with a non-linear activation function, rather than an embedding layer, due to the encoding method of the PoS features, as explained in Section 4.1. For the mini-batch size, the authors recommend using the batch size between 8 and 32, depending on the size of the training dataset. However, that is the result on the CoNLL-2003 dataset, which is an English dataset. We use 50 similarly to the German NER model in (Reimers and Gurevych, 2017).

We should mention that the CRF layer implemented by the *Keras-contrib* package offers two different modes for the training and testing processes: "Join" and "Marginal" for training, "Viterbi" and "Marginal" for testing. The "Join" training mode and "Viterbi" testing mode follows the "vanilla" fitting algorithms for linear chain CRF. "Marginal" training is optimized via composition likelihood (product of marginal likelihood), which is not optimal in this case. "Marginal" testing mode will decode the input sequence according to the training result and compute marginal probabilities. In this mode, it can therefore output a probability prediction of the classes for tokens. According to the documentation, the "Join" training mode can outperform the other training mode, and the "Viterbi" testing mode can achieve better performance than "Marginal" testing mode, but reasonably close. In this work, we evaluate using both "Join-Marginal" and "Join-Viterbi" modes.

5 Performance and evaluation

In this section, we report the performance for the automatic projection pipeline and the NER model. F1-score is used as the evaluation metric. The overall F1-score is the weighted average F1-score of each category.

5.1 Automatic projection pipeline

We currently utilize data only from the beginning of 2017 to July of 2018 for development, model training and validation. The total Finnish data consists of around 83,000 articles.

Our English data, on the other hand, date back 6 years from 2018 to 2012. Only articles from the same time period as the Finnish data can be used for name matching. The amount of usable English articles is around 4,486,000. We consider the NER performance only on "person", "location" and "organization" tags, to make the final outcome comparable to the Polyglot Finnish NER tagger.

To evaluate the performance of the automatic projection, we manually checked 1,000 randomly selected sentences from March 2018 to April 2018. Since our three English NER taggers have different

Tag	*Prec*	*Rec*	*F1*	*Support*
B-PER	0.97	0.99	0.98	823
I-PER	0.97	0.97	0.97	668
B-LOC	0.99	0.99	0.99	341
I-LOC	1.00	0.67	0.80	3
B-ORG	0.99	0.98	0.98	536
I-ORG	1.00	0.82	0.90	78
Avg / total	0.98	0.98	0.98	2449

Table 3: Quality of Finnish data projected from PULS pattern-based NER: evaluated on 1,000 sentences, annotated manually.

Tag	*Prec*	*Rec*	*F1*	*Support*
B-PER	0.99	0.97	0.98	776
I-PER	0.99	0.97	0.98	639
B-LOC	0.97	0.97	0.97	376
I-LOC	0.55	0.60	0.57	10
B-ORG	0.95	0.98	0.96	587
I-ORG	0.91	0.87	0.89	92
Avg / total	0.97	0.97	0.97	2478

Table 4: Quality of Finnish data projected from BiLSTM-CRF-W2V model: evaluated on 1,000 sentences, annotated manually.

performance, the manual evaluation is conducted separately, as shown in Table 3, Table 4 and Table 5.

5.2 NER model

For training the NER models, we used data from 2017-01 to 2017-12 (12 months). This period contains 50,009 Finnish documents, for which we found 920,658 matching English documents. We filtered out projected sentences for which the English tagger produced NER tags *other than* Person, Organization, or Location. This data produced approximately 114,000 automatically projected sentences after filtering. For validation, we used two months: 2018-04 to 2018-05. This period contained 11,452 Finnish documents, which had 389,072 English matching documents. This data produced 23,277 automatically projected sentences, after filtering.

Instances are projected from 3 different English NER taggers: the PULS pattern-based tagger, BiLSTM-CRF-GloVe tagger and BiLSTM-CRF-W2V tagger. Two train-test modes, “Join-Marginal” and “Join-Viterbi”, are also applied for comparison. Six different Finnish NER models are evaluated.

Tag	*Prec*	*Rec*	*F1*	*Support*
B-PER	0.96	0.99	0.97	767
I-PER	0.97	0.97	0.97	632
B-LOC	0.97	0.96	0.96	403
I-LOC	0.71	0.53	0.61	19
B-ORG	0.96	0.97	0.96	639
I-ORG	0.97	0.78	0.87	101
Avg / total	0.96	0.96	0.96	2561

Table 5: Quality of Finnish data projected from the BiLSTM-CRF-GloVe model: evaluated on 1,000 sentences, annotated manually.

Table 6 shows the model evaluation on this data.

As expected, upon visual inspection, we noticed instances with incorrect “ground truth” since projection is not entirely clean. Despite its good overall quality, the validation performance may still differ from actual performance.

We conducted further model testing and inspection to obtain better estimates of the true performance. We sampled another set of articles from 2018-08 to 2018-10 (3 months), which is outside our automatic projection time period. For further inspection and error analysis, in Section 5.3, we randomly sampled a total of 36 articles, evenly from the following 6 sections of the newspaper:

- “Talous” (Economics)
- “Politiikka” (Politics)
- “Ulkomaat” (Foreign news)
- “Kotimaa” (Domestic news)
- “Koti” (Home)
- “Kaupunki” (The City)

The first three of these categories are more closely related to the Business domain. Again, articles are **evaluated manually**. The result is shown in Table 7. Polyglot is used as the performance baseline. Two additional Finnish NER models are trained with full FiNER-data (Ruokolainen et al., 2019), including the validation and test dataset, for better comparison.[5]

As shown in Table 7, Polyglot and the model trained with FiNER-data have better precision than most of our Finnish NER taggers, but have a worse recall rate. Overall, most of our Finnish NER taggers achieve better performance. Only one model, projected from BiLSTM-CRF-W2V in Join-Viterbi mode, performs worse than the FiNER-data model

[5] `www.github.com/mpsilfve/finer-data`

Eng-NER Source	*Train-test mode*	*Prec*	*Rec*	*F1*	*Support*
PULS pattern-based	Join-Viterbi	0.94	0.92	0.93	28858
PULS pattern-based	Join-Marginal	**0.94**	**0.93**	**0.93**	28858
BiLSTM-CRF-GloVe	Join-Viterbi	0.92	0.89	0.90	34526
BiLSTM-CRF-GloVe	Join-Marginal	0.93	0.91	0.92	34526
BiLSTM-CRF-W2V	Join-Viterbi	0.87	0.83	0.84	37219
BiLSTM-CRF-W2V	Join-Marginal	0.87	0.85	0.85	37219

Table 6: Validation scores on 2018-04 to 2018-05. "PULS pattern-based" and "BiLSTM-CRF-*" refer to the Finnish NER models that are projected from our PULS pattern-based NER tagger and English BiLSTM-CRF NER tagger respectively. "GloVe" and "W2V" indicates the embedding that English NER taggers use.

NER Source	*Train-test mode*	*Prec*	*Rec*	*F1*	*Support*
PULS pattern-based	Join-Viterbi	**0.89**	0.77	**0.82**	916
PULS pattern-based	Join-Marginal	0.80	**0.83**	0.81	916
BiLSTM-CRF-GloVe	Join-Viterbi	0.80	0.75	0.76	916
BiLSTM-CRF-GloVe	Join-Marginal	0.79	0.74	0.75	916
BiLSTM-CRF-W2V	Join-Viterbi	0.76	0.72	0.73	916
BiLSTM-CRF-W2V	Join-Marginal	0.78	0.79	0.78	916
FiNER-data	Join-Viterbi	0.83	0.72	0.75	916
FiNER-data	Join-Marginal	0.73	0.68	0.64	916
Polyglot		0.82	0.55	0.64	916

Table 7: Test evaluation. "FiNER-data" refer to the Finnish NER model trained with data from FiNER-data. "Polyglot" entry illustrates the performance of their model on our test dataset

in Join-Viterbi mode. The Finnish NER tagger that is projected from the PULS pattern-based English NER tagger in Join-Viterbi mode achieves the overall best performance. This result also suggests that the dataset produced by our automatic projection pipeline is valid for model training, data of large size and various topics, while FiNER-data only covers technology-related news.

5.3 Error analysis

Despite good overall performance on all automatically projected datasets, the average F1 score of models with the different setups is still around 77%. More work is required to improve performance. To guide future work, we did further visual inspections to examine the predictions in general.

One major problem is that the NER model gets data from automatic projection with **limited pattern diversity**. During the training pipeline, including automatic projection, there are two reasons that may cause this problem.

Firstly, flaws still exist in the automatic projection pipeline. One flaw that shows up often during visual inspection is that the projection currently does not support named entities without any capital letters. Named entities such as "Valkoinen talo" (the White House) cannot be fully recognized at the beginning of the pipeline (because the second token is lowercase). As a result, only the token with a capitalized letter such as "Valkoinen" will be predicted as a named entity by Finnish NER tagger.

Secondly, the English data source is biased in favor of foreign business topics. Within the time period of the training data, our database contains mostly business news. As a consequence, more business-related news and their named entities in Finnish news can be tagged. General news may behave differently than business news, and may contain more patterns for the task of NER. To verify this conjecture, we inspect each category with the model projected from the PULS pattern-based tagger in Join-Viterbi mode. As shown in Table 8, the model can achieve better performance on average on the topics which are related to foreign business or politics news, compared to domestic or local news.

Another major problem is due to a **flaw in**

Category	*Prec*	*Rec*	*F1*	*Support*
Overall	0.89	0.77	0.82	916
Talous	0.91	0.90	0.90	123
Politiikka	0.89	0.79	0.83	191
Ulkomaat	0.92	0.75	0.83	227
Kotimaa	0.90	0.70	0.78	100
Koti	0.86	0.73	0.77	167
Kaupunki	0.83	0.71	0.74	108

Table 8: Performance of the Finnish NER tagger for each category. The tagger is projected from the pattern-based English NER tagger in Join-Viterbi mode, first line in Table 7.

Tag	*Prec*	*Rec*	*F1*	*Support*
B-PER	0.88	0.70	0.78	20
I-PER	0.83	1.00	0.91	10
B-LOC	0.85	0.90	0.88	52
I-LOC	0.00	0.00	0.00	5
B-ORG	1.00	0.32	0.48	19
I-ORG	0.00	0.00	0.00	2
avg/total	0.83	0.71	0.74	108

Table 9: Detailed performance of Finnish NER tagger for category “Kaupunki” (“The City”) in Table 8.

data encoding. As mentioned previously, out-of-vocabulary compound lemmas are decomposed and assigned the embedding of the last part of the compound lemma. Many ordinary tokens may benefit from this approach, while organization named entities do not. During visual inspection, we noticed that the name of some Finnish national or local governmental departments can be a made-up word or a compound word. Such names will either be assigned the “UNKNOWN” token embedding or the embedding of the last part of the compound word, which is most likely a common noun. For example, “Verohallinto” (Tax Administration) does not have an embedding as a whole. However, “hallinto” (“government”) is a common noun within the embedding vocabulary. As a result, these named entities are more likely to be tagged incorrectly. As illustrated in Table 9, the performance of organizations (B-ORG) suffers from severely low recall rates due to this problem, as well as the previously mentioned problem that Finnish domestic named entities are less likely to get projections in the pipeline.[6]

6 Conclusions and future work

In this paper, we propose the idea of building a Finnish NER dataset by leveraging the output of an English NER tagger and projecting the type of recognized named entities from English to Finnish. The contributions of this paper are:

- Our work shows that the Finnish NER dataset produced by only simple rule-based projection can be used for NER model training. No parallel bilingual documents are used, only projected named entities, obtained by several *monolingual* tools.
- We demonstrate the performance of our NER model, and set a new benchmark for Finnish NER.

For future work, we plan to first tackle the problems that we mentioned in the error analysis section and conduct further inspection. Secondly, we plan to combine our pipeline with a disambiguation model, to improve both the pre-processing and the data encoding steps. Thirdly, it would be interesting to experiment and generalize our approach with other languages with limited NER tools, such as Estonian, if corresponding news datasets are easily accessible.

Acknowledgements

This work was supported in part by TEKES, the Finnish Funding Agency for Technology and Innovation, Project “iMEDL—Digital Media Evolution through Deep Learning”, number 5933/31/2017.

References

Rami Al-Rfou, Vivek Kulkarni, Bryan Perozzi, and Steven Skiena. 2015. Polyglot-ner: Massive multilingual named entity recognition. In *Proceedings of the 2015 SIAM International Conference on Data Mining*. SIAM, pages 586–594.

Jason P.C. Chiu and Eric Nichols. 2016. Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics* 4:357–370.

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

[6] Because they are unlikely to appear in English-language news.

Mian Du, Lidia Pivovarova, and Roman Yangarber. 2016. PULS: natural language processing for business intelligence. In *Proceedings of the 2016 Workshop on Human Language Technology*.

Maud Ehrmann, Marco Turchi, and Ralf Steinberger. 2011. 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.

Jenny Rose Finkel, Trond Grenager, and Christopher Manning. 2005. Incorporating non-local information into information extraction systems by Gibbs sampling. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Stroudsburg, PA, USA, ACL '05, pages 363–370.

Abbas Ghaddar and Phillippe Langlais. 2017. Winer: A Wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. pages 413–422.

Katri Haverinen, Jenna Nyblom, Timo Viljanen, Veronika Laippala, Samuel Kohonen, Anna Missilä, Stina Ojala, Tapio Salakoski, and Filip Ginter. 2014. Building the essential resources for Finnish: the Turku Dependency Treebank. *Language Resources and Evaluation* 48(3):493–531.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *CoRR* abs/1508.01991.

Jun'ichi Kazama and Kentaro Torisawa. 2007. Exploiting Wikipedia as external knowledge for named entity recognition. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*.

Sungchul Kim, Kristina Toutanova, and Hwanjo Yu. 2012. Multilingual named entity recognition using parallel data and metadata from Wikipedia. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, pages 694–702.

Veronika Laippala and Filip Ginter. 2014. Syntactic N-gram collection from a large-scale corpus of internet Finnish. In *Human Language Technologies-The Baltic Perspective: Proceedings of the Sixth International Conference Baltic HLT 2014*. IOS Press, volume 268, page 184.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Association for Computational Linguistics, San Diego, California, pages 260–270.

Xuezhe Ma and Eduard H. Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. *CoRR* abs/1603.01354.

Tomas Mikolov, Kai Chen, Greg S Corrado, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. In *NIPS*.

Sjur N Moshagen, Tommi Pirinen, and Trond Trosterud. 2013. Building an open-source development infrastructure for language technology projects. In *Proceedings of NODALIDA 2013: the 19th Nordic Conference of Computational Linguistics*. Oslo, Norway.

Joel Nothman, Nicky Ringland, Will Radford, Tara Murphy, and James R Curran. 2013. Learning multilingual named entity recognition from Wikipedia. *Artificial Intelligence* 194:151–175.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proc. of NAACL*.

Nils Reimers and Iryna Gurevych. 2017. Optimal hyperparameters for deep lstm-networks for sequence labeling tasks. *CoRR* abs/1707.06799.

Alexander E Richman and Patrick Schone. 2008. Mining wiki resources for multilingual named entity recognition. In *Proceedings of ACL-08: HLT*. pages 1–9.

Teemu Ruokolainen, Pekka Kauppinen, Miikka Silfverberg, and Krister Lindén. 2019. A finnish news corpus for named entity recognition. *Language Resources and Evaluation* pages 1–26.

Antonio Toral and Rafael Munoz. 2006. A proposal to automatically build and maintain gazetteers for Named Entity Recognition by using Wikipedia. In *Proceedings of the Workshop on NEW TEXT Wikis and blogs and other dynamic text sources*.