UPPSALA
UNIVERSITET

Named-entity recognition in Czech historical texts

Using a CNN-BiLSTM neural network model

Helena Hubková

Uppsala University
Department of Linguistics and Philology
Master Programme in Language Technology
Master's Thesis in Language Technology, 30 ECTS credits

June 15, 2019

Supervisors:
Eva Pettersson, Ph.D., Uppsala University
doc. Ing. Pavel Král, Ph.D., University of West Bohemia

Abstract

The thesis presents named-entity recognition in Czech historical newspapers from *Modern Access to Historical Sources Project*. Our goal was to create a specific corpus and annotation manual for the project and evaluate neural networks methods for named-entity recognition within the task.

We created the corpus using scanned Czech historical newspapers. The scanned pages were converted to digitize text by optical character recognition (OCR) method. The data were preprocessed by deleting some OCR errors. We also defined specific named entities types for our task and created an annotation manual with examples for the project. Based on that, we annotated the final corpus.

To find the most suitable neural networks model for our task, we experimented with different neural networks architectures, namely long short-term memory (LSTM), bidirectional LSTM and CNN-BiLSTM models. Moreover, we experimented with randomly initialized word embeddings that were trained during the training process and pretrained word embeddings for contemporary Czech published as open source by fastText[1]. We achieved the best result F1 score 0.444 using CNN-BiLSTM model and the pretrained word embeddings by fastText. We found out that we do not need to normalize spelling of our historical texts to get closer to contemporary language if we use the neural network model. We provided a qualitative analysis of observed linguistics phenomena as well. We found out that some word forms and pair of words which were not frequent in our training data set were miss-tagged or not tagged at all. Based on that, we can say that larger data sets could improve the results.

[1] https://fasttext.cc/

Contents

Preface

I want to thank Eva Pettersson for her thorough feedback, Pavel Král from University of West Bohemia for help and valuable advises and Modern Access to Historical Sources Project for providing of data.

1. Introduction

Named-entity recognition (NER) is one of the basic tasks in natural language processing (NLP). NER has a goal to automatically find and identify entities of interest in the texts, e.g. personal names, time expressions, names of institutions or geographical names.

The main goal of the thesis is to automatically detect and classify named-entities (NEs) in Czech historical texts. Our sources are texts from *Modern Access to Historical Sources Project* which are presented through the portal Porta Fontium[1]. One of the goals of the Project is to create an intelligent full-text access to the printed documents from archival resources from the Czech-Bavarian border region.

In our thesis, we want to explore if using a state-of-the-art neural network (NN) model for NER can be used in case of Czech historical data and find the most suitable NN architecture for our task. We want to also analyze if the NN model is able to skip a preprocessing step of spelling normalization in NER task using historical data.

In the first part of the thesis project, we create and annotate a corpus. The thesis has a focus on the Czech historical texts from newspapers called *Posel od Čerchova* from the second half of the 19th century. The texts are processed by optical character recognition (OCR) at first and OCR errors are manually corrected. We define specific NE based on the project purpose itself in the combination with named-entities types from Ševcikova et al. (2007) and we create a manual to annotate the corpus.

In the second part of the thesis project, we evaluate different neural networks architectures for our task, concretely, recurrent neural long short-term memory (LSTM) network, bidirectional LSTM (BiLSTM) and BiLSTM in combination with convolutional neural network (CNN) architecture to modeling character-level features. Our models are built based on a model by Chiu and Nichols (2016). We also experiment with two different approaches to word embeddings. First, we implement empty vectors and we train them during the experiments from their initial point, secondly, we use published pretrained word embeddings for contemporary Czech provided by fastText[2].

We evaluate our models using standard metrics - *precision*, *recall* and *F1-score*. We compare plain text of evaluation test set against its manually annotated gold standard version. We also optimize hyper-parameter settings for our best model and we evaluate the final chosen model using these settings. We also provide qualitative analysis of output tagged text to explore which linguistic phenomena of historical input data sets caused problems for automatic detection of NE and NE tagging itself.

[1] http://www.portafontium.eu/
[2] https://fasttext.cc/

More details about the background of our work can be found in Chapter 2. In Chapter 3, we describe our data and the annotation process. We explain our methodology and neural networks models in Chapter 4. The evaluation and results can be found in Chapter 5. Conclusion of our work and ideas for the future work are presented in Chapter 6 at the end of the thesis.

2. Background

2.1. Natural language processing for historical texts

Natural language processing of historical texts can be challenging because the historical text usually has a high level of spelling differences (it differs from the contemporary orthography), variance (inconsistency) and uncertainty (digitization errors etc.). This can be problematic for testing different NLP tools and techniques (tokenization, POS-tagging, NER etc.) for processing the historical texts. (Piotrowski, 2012a)

Nowadays, the usual approach is project-specific spelling normalization of the historical texts to make their spelling closer to a modern language text. It means that in case that particular word spelling (or word form) is not used in contemporary language, a token (or a sequence of tokens) from a historical text is mapped to a token with contemporary spelling. This step enable to use same NLP tools as for modern language texts. (Piotrowski, 2012b)

For example, the Czech historical sentence *Na tyto otázky snažiti se budu dáti co možná uspokojující odpověd.* ("I will try to answer these questions as satisfactory as possible.") could be normalized to *Na tyto otázky snažit se budu dát co možná uspokojující odpověď.*

There are different spelling normalization approaches ranging from rule-based methods, edit distance approaches, statistical machine translation (SMT) to neural machine translation (NMT). For example, Korchagina (2017) compared three approaches to spelling normalization for German historical texts from 15th–16th centuries: rule-based, SMT, and NMT.

However, some recent research used neural networks for processing historical texts the way which allows to skip the preprocessing step - the spelling normalization. For instance, Hardmeier (2016) described the method using character-based recurrent neural network (RNN) for a part-of-speech tagger for historical Swedish and German texts.

2.2. Named-entity recognition in general

Named-entity recognition means finding and identifying entities of interest in the texts. The entities of interest (words or sequences of words) can be personal names, names of organizations, locations, time expressions, numerical expressions etc. What is defined as a NE depends on the purpose of the text processing. Generally, NER is part of many NLP tasks such as information retrieval, machine translation, question answering, sentiment analysis etc. (Jurafsky and Martin, 2009)

In case of methodology, rule-based approaches used to be preferred, however, supervised and semi-supervised machine learning methods and their combinations

are usually used for NER today. (Chiticariu et al., 2013) Supervised machine learning method needs only annotated data (corpora), on the other hand, semi-supervised methods work not only with annotated data but also with other unlabeled data. Recently, the most used methods are conditional random fields (CRF), maximum entropy recognition and approaches using different neural networks models (and their combinations).

2.3. Named-entity recognition and the Czech language

NER in Czech has a quite long tradition in terms of data for the contemporary Czech language. Ševčíková et al. (2007) introduced two-level classification and used that for manually annotating 11 000 NEs. Based on that, they developed a Czech NE tagger. They distinguished NE *span recognition* (all NEs are found but the type is not relevant), NE *supertype recognition* (all NEs are found and supertype - first-level - is correct) and NE *type recognition* (all NEs are found and both supertype and type - second-level - are correct). They evaluated the tagger using precision, recall and F-measure metrics. For all NE instances, they got precision 74%, recall 54% and F-measure 62% in case of correct type, then precision 81%, recall 59% and F-measure 68% in case of correct supertype and finally, precision 88%, recall 64% and F-measure 75% in case of correct span.

Similarly, Kravalova and Zabokrtsky (2009) presented the Czech NE corpus (CNEC) which used the two-level classification scheme and it consists of around 6 000 sentences. They also used a Support Vector Machine classification approach for training and evaluating data for NER. They distinguished NEs according to Ševčíková et al. (2007). They got precision 75%, recall 62% and F-measure 68% for type recognition (span and type). In case of correct span and supertype, they achieved precision 75%, recall 67% and F-measure 71%. However, span recognition itself performed precision 84%, recall 70% and F-measure 76%.

Moreover, Král (2011) created a NER system for Czech News Agency to evaluate different features for NER to find an "optimal" set of features. They classified the system using Conditional Random Fields (CRFs) and tested using Czech NER corpus (Kravalova and Zabokrtsky, 2009). They achieved 58% of F-measure with the best feature set.

Straková et al. (2013) built a NE recognizer based on Maximum Entropy Markov Model and a Viterbi algorithm and evaluated it for Czech and English. They achieved 82.82% F-measure for Czech using Czech Named Entity Corpus 1.0 and F-measure 89.16% for English using CoNLL-2003.

Similarly, Straková et al. (2014) presented two open-source taggers: NER tagger *NameTag* and *MorphoDiTa* (Morphological Dictionary and Tagger) for morphological analysis. Both tools are specifically designed for inflective languages including Czech.

2.4. Named-entity recognition in historical texts

The issue of NER in historical texts has been described by several researchers so far. Grover et al. (2008) built a rule-based NER system recognizing names of places and persons in digitized records of British parliamentary proceedings

texts from two different periods, the late 17th and early 19th centuries. They focused on issues caused by the nature of historical texts (a high level of variance in using word-initial capitals) and by using optical character recognition (OCR) technology. They found out that recognition of personal names achieved better results than recognition of names of places. In other words, finding patterns for personal names recognition was easier and therefore more resistant to OCR errors. On the other hand, they also described other problems caused by OCR errors: wrong interpretation of layout, wrong division of tokens and wrong division of paragraphs (in the middle of a token). They reached F1 score 71.81% for the period 1814-1817 and 70.35% for the period 1685-1691.

Packer et al. (2010) experimented with recognition of person names using noisy OCRed data. They tried three different approaches and evaluated the output against hand-labeled test data. They showed that the character-level errors in OCRed data have small impact to NER in comparison to word order errors.

Rodriguez et al. (2012) evaluated four different tools for NER in historical texts: a) OpenNLP , b) Stanford NER (Finkel et al., 2005), c) AlchemyAPI , d) OpenCalais. They used EHRI project[1] documents as input and defined three NE - *person*, *location* and *organization*. They showed that the Standford NER has overall best performance (especially in case of person and location entities), similarly, Alchemy API worked the best for NE type organization in case of manually corrected text. OpenNLP showed the lowest accuracy.

Mac Kim and Cassidy (2015) applied the Stanford NER system to the 155 million OCRed articles from historical Australian newspapers to recognize NE types *person*, *location* and *organization* and they showed how the data can be exploited using a clustering method.

Neudecker (2016) created an open corpus for NER in Dutch, French and German OCRed historical newspapers. The work was included in the Europeana Newspapers project[2]. They used Stanford NER for preprocessing German data, otherwise, they annotated NE in data manually. They distinguished only NEs *person*, *location* and *organization* in the corpus.

Won et al. (2018) evaluated NER tools (NER-Tagger, Stanford NER, Edinburgh Geoparser, Spacy, Polyglot and their combinations) based on the ability to automatically identify place names in historical correspondence. They showed that the NER systems are corpus dependent and, moreover, preprocessing and translation to Modern English does not affect NER results significantly.

The area of NER in *Czech historical texts* is not so explored yet. Hana et al. (2011) developed a morphological tagger for Old Czech (1200-1500 AD). They used transformation of Modern Czech to Old Czech and vice versa. They "aged" the tagger for Modern Czech and used it for tagging the modernized corpus of Old Czech. Then they replaced the modernized word-forms with original word-forms and this resulting tagged corpus was the final training corpus for developing the final Old Czech Tagger. The advantage of this approach is that they minimized amount of language-specific work, however, they got worse results (only 74%) than the results of a traditional tagger such as 94% obtained by Morče tagger[3].

[1] http://www.ehri-project.eu
[2] http://www.europeana-newspapers.eu/
[3] http://ufal.mff.cuni.cz/morce/index.php

2.5. Named-entity recognition using neural networks

Experiments with neural networks are quite common in NER research nowadays. For example, Collobert et al. (2011) presented the unified multilayer convolutional neural network (CNN) model with a learning algorithm which can be used in various NLP tasks including NER. They experimented with training data which were mostly unlabeled and not optimized for each NLP task. For the NER task, they achieved F1 score 81.47% for random category (embedding vectors are initialized randomly) and 89.59% for Senna category (using Senna word-embeddings).

Huang et al. (2015) compared different Long Short-Term Memory (LSTM) approaches for sequence tagging. They worked with bidirectional LSTM (BI-LSTM) networks, LSTM with a Conditional Random Field (CRF) layer and bidirectional LSTM with a CRF layer (BI-LSTM-CRF). They showed that using both past and future input in bidirectional component of BI-LSTM-CRF is efficient and, also, that the CRF layer of the model helps by using sentence level tag information. The system achieved state-of-art accuracy results in term of POS, chunking and NER data sets. They compare they results with Collobert et al. (2011)and they got the F1 score 84.26% for random initialized vector embeddings and 90.10% for NER using Senna word-embeddings.

Also, Chiu and Nichols (2016) built a hybrid bidirectional LSTM and CNN model which automatically detects character-level and word-level features. They showed that the system has similar performance to the CoNLL-2003 data set and, moreover, the performance is 2.13 F1 points better than previous research using OntoNotes 5.0. They achieved F1 score 91.62% on CoNLL-2003 and 86.28% on OntoNotes.

Finally, Lample et al. (2016) introduced two neural models - bidirectional LSTM CRF and a transition-based model using shift-reduce parsers. For their experiments, they used character-based word representations based on the supervised corpus and unsupervised word representations based on the unannotated corpora. Both models achieved better results than in the previous research including models using external resources (e.g. gazetteers), concretely, the LSTM CRF model reached F1 score 90.94% for English NER, F1 score 78.76% for German, 81.74% for Dutch and 85.75% for Spanish using labeled training external data. In case of English NER, the LSTM CRF model which was pretrained by word embeddings, includes character-based modeling of words and dropout rate achieved F1 score 90.94%.

3. Data

For our experiments, we used Czech historical texts from the *Modern Access to Historical Sources Project*[1], concretely, we worked with Czech historical newspapers called *Posel od Čerchova*. These newspapers were published from 1872-1935, however, only scans of issues till 1900 are available on the Porta Fontium portal so far. For our corpus, we chose first 32 issues from 1872 for further annotations. This number of issues guaranteed more than 70 000 tokens for our final corpus in this thesis, however, there are plans to build bigger corpus for future experiments within the project which should cover also issues from other years.

3.1. Comparison of our data to contemporary Czech

Our data are publicists texts from 1872 and they differ especially in vocabulary, word forms, spelling and word order in comparison to contemporary newspapers texts. In term of vocabulary, the texts contain archaic words from 19th century which are more or less understandable to a contemporary reader. For example, word *an* which can be used in the sense of *which* or *when* based on the context (in Czech *který*, *když*, respectively), word *údové* which means "members" (cotemporary Czech: *členové*) or word *vůkol* which means "around" (*okolo*).

In case of spelling, we can find differences that would be considered a spelling mistake today (e.g. *výtězně*, contemporary spelling: *vítězně*, "triumphantly"; *ouklady*, contemporary: *úklady*, "machinations"; *věčší*, contemporary *větší*, "bigger").

A relatively large group of differences between Czech used in our data and contemporary Czech are differences in word forms. Archaic word forms of the verb "to be" such as *býti* (infinitive), *jest* (3. person singular) or *jsouť* (at the beginning of a sentence in the form of a particle) are used regularly in the texts. Similarly, verb form for infinitive ending with *ti* is the only one that appears in the texts (e.g. *chrániti* - "to protect", *docíliti* - "to achieve" or *pěstovati* - "to cultivate") in comparison to contemporary regular ending for verbs *t* (contemporary: *chránit*, *docílit* or *pěstovat*, respectively). Moreover, transgressive verb forms which are also considered as archaic occur more often in our texts than in contemporary ones. For instance, *sestoupivše do spolku* - "joined the association", *vynášeje* - "bringing out" or *jdouc* - "going".

We can find not only different word forms of verbs but also of nouns: *občanstvo* - "citizens" (in comparison to *občané*), *zástupcové* - "representatives" (*zástupci*); adjectives: *žádoucno* - "desirable" (in comparison to *žádoucí*) and pronouns *všickni* -"everyone" (*všichni*) and *kteráž* - "who" (*která*).

[1] http://www.portafontium.eu/

Finally, if we compare word order of our newspaper texts with contemporary ones, we can say that nowadays word order regularly uses adjectives as premodifiers e. g.: *německá říše* - "German Empire". However, in our texts, we can find these adjectives more often as postmodifiers, e.g.: *říše německá*. Also, the position of the predicate (*byly vyschlé*) is usually at the end of a sentence in our texts (*dodržujícím parnem jako troud vyschlé byly* - "they were dried up to cinder by steady steam"), but nowadays, we would rather write *dodržujícím parnem byly vyschlé jako troud*.

In addition to examples above, we can also find archaic abbreviations in our texts. They include different time expressions e.g. *14. t. m.* in full words *14. tohoto měsíce* which means "14th of this month", similar *t. r.* in full words *tohoto roku* - "this year". A specific abbreviation is also *pp.* which is the plural form of the abbreviation *p.* ("Mr.") and *c. k.* which means *císařsko-královský* ("imperial-royal") which was used in titles of organizations in Austrian part of Austria-Hungary in the second half of the 19th century. These abbreviations are usually part of NEs and therefore could be problematic to automatically detect.

Moreover, we have to mention that the texts contain also words and sentences which are not in Czech. There are also some quotations in German and Latin, however, the amount of these words (sentences) is negligible.

3.2. Preprocessing of data

In the first step, we used optical character recognition (OCR), concretely tesseract 4.0.0.[2], to transfer chosen scanned documents to digitized plain texts. The output of this step includes noisy data which caused the most problematic parts of OCRed output data, therefore we do not consider a few pages which were badly scanned and manually corrected evident noisy parts.

The next step includes automatic deleting non-existing characters and other symbols which are never used in original texts: €, $, £, #, *, +, =, ©, <, >, |, [,], » and «. We also joined words that were separated at the end of a line and we automatically tokenized the text. The preprocessed data prepared for annotations had 75,200 tokens. However, it was hard to automatically correct some OCR errors. For example, there are many dots and commas (and other characters such a dash, question mark or exclamation mark) which are only noise in OCRed data therefore it is difficult to find the end of sentence.

3.3. Defining entities

Before we annotated data for our corpus, we defined our own types of entities for our project. The NE types were inspired by Czech Named Entity Corpus (CNEC) (Kravalova and Zabokrtsky, 2009) and their NE classification system, however, we focused also on the purpose of the project and the nature of the historical texts.

[2]https://github.com/tesseract-ocr/tesseract

Table 3.1.: Defined named-entities and what they include

Named-entity	Tag	**NE includes:**
Personal names	**p**	first names, surnames, artistic names, (academic) titels, (royal) family names
Institutions	**i**	names of institutions, organizations, clubs, companies, names of historical collectives (e. g. religious orders)
Geographical names	**g**	names of continents, states, territorial-administrative units, streets and public places, natural monuments including local names
Time expressions	**t**	date, days, hours, month, years, centuries, names of epochs, holidays and important days, historic events
Artifact names	**o**	names of documents, artworks, products, books, newspapers, buildings, currency
Ambiguous	**a**	used in case the annotator is not sure which of the types above is correct

CNEC presented a two-level annotation system which has 10 basic types of entities in the first level (*Numbers in addresses*, *Bibliographic items*, *Geographical names*, *Institutions*, *Media names*, *Specific number usages*, *Artifact names*, *Personal names*, *Quantitative expressions*, *Time expressions*).

Table 3.2.: Example from the final corpus

zvolený	UNK	O	CS
ústavácký	UNK	B-i	CS
zemský	UNK	I-i	CS
výbor	UNK	I-i	CS
zastaví	UNK	O	CS
stavbu	UNK	O	CS
české	UNK	B-i	CS
polytechniky	UNK	I-i	CS
,	UNK	O	CS

On the other hand, previous research in NER for historical texts usually used only three types of entities: *person*, *location* and *organization* (e.g. Rodriguez et al. (2012), Mac Kim and Cassidy (2015) and Neudecker (2016). More in Section 2.4.).

Our texts are from historical newspapers from the 19th century and therefore we can say that the structure of the newspapers is similar to contemporary ones (articles about news from home, from abroad, advertisements etc.). However, some of the entities types according to CNEC are not common in our texts or are not important for future usage of named-entities in our project (*Numbers in addresses*, *Specific number usage*, *Quantitative expressions*). In case of CNEC NE type *Media names*, we included names of media into one of our types (*Artifact names*, more below). Similarly, almost all subtypes of the CNEC NE type *Bibliographic items - page numbers*, *volume numbers* etc. are included in the metadata of our scanned texts.

Based on what has been said above, we defined five basic types of NEs for Czech historical newspapers: *Personal names*: **p**, *Institutions*: **i**, *Geographical*

Table 3.3.: Numbers of annotated tokens by each of the annotators divided into NE types

NE type	Tag	Annotator 1	Annotator 2	Final
Personal names	p	2683	2613	2087
Institutions	i	1152	1178	1061
Geographical names	g	923	933	728
Time expressions	t	1208	1071	892
Artifact names (objects)	o	1644	1621	1436
Ambiguous	a	2160	2078	1890
All		9770	9494	8094

names: **g**, *Time expressions*: **t**, *Artifact names* (or *Objects*: **o** and one specific NE type *Ambiguous*: **a** which helps annotators if the entity type is not clear.

To sum up, we created a basic level of NE type classification which can be extended by other level in the future. In Table 3.1., we can see the named-entities and description of sub-type names included. A detailed annotation manual with examples can be seen in **Appendix** of the thesis.

3.4. Data annotation

We manually annotated preprocessed data using the entities described in Section 3.3. We decided that the best format for our final corpus will be modified conll format[3]. That format is easy to read and it is also easy to change the information in corpus for future needs. As can be seen in Table 3.2., each line of corpus contains four columns. In the first one, there is one token and the second one is the space for lemma. In our case, we filled default "UNK" option as unknown lemma. That could be changed in the future if there is a need to consider lemmatization. At the third position, there is our NE type tag ("p", "i", "g", "t", "o" and "a"). We

Table 3.4.: Numbers of tagged tokens and NEs of each NE type in our corpus

NE type	Tag	Number of tagged tokens	Number of NEs
Personal names	p	2087	661
Institutions	i	1061	131
Geographical names	g	728	207
Time expressions	t	1436	314
Artifact names	o	892	170
Ambiguous	a	1890	202
All		8094	1685

also used "IOB" tags to indicate the first word in a multiword entity (tag "B" as "beginning"), and inside words of all other NEs (tag "I" as "internal"). All tokens that are not a NE are tagged as "O" - "outside" (Jurafsky and Martin, 2009). In our example in Table 3.2., we can see a name of an institution *ústavácký zemský výbor* ("constitutional committee") with the tags in the third column *B-i*, *I-i* and *I-i*.

[3]https://www.clips.uantwerpen.be/conll2003/ner/

Finally, the last column contains information about language. Most tokens are Czech ("CS"), however, we can find some German ("DE") and Latin ("LA") tokens in the corpus.

All the data were manually annotated by two annotators and we decided to work only with the intersection of their annotations to avoid data which could be considered as ambiguous. We can see an overview of tagged tokens of each NE type by each of the annotators in Table 3.3.. Annotators agreed on 8,094 tagged tokens and disagreed in 3,076 cases that were not included into our corpus. For the future work within the project, we plan to review the data sets again and add another annotator.

Finally, each sentence in corpus is separated by an empty line. Division of corpus into sentences is crucial for using of our NN model. However, because of the noisiness of our data caused by OCRed pre-processing, this step became quite complicated. We tried to automatically divided sentences based on a rule, that the end of sentence is after a dot, a question mark or an exclamation mark and before a capital letter. This rule cannot be applied for all sentences (especially problematic are abbreviations in a middle of sentence). In the future work, this weakness of our data sets should be revised and manually corrected.

Our final corpus includes 1685 NEs, both one token entities and NE consists of more than one token. It means that 8094 tokens from our corpus are tagged with one of our 6 NE type tags. An overview of numbers of NEs and tagged tokens for each NE type can be seen in Table 3.4.

NE type *Personal names* consists of 661 NE (and 2087 tokens) and it is the largest NE type in our corpus, on the other hand, only 207 of *Geographical names* (728 tokens) is in our data sets.

3.5. Data division

Finally, the annotated corpus (75,200 tokens) was divided into three sets. We used 80% of corpus for training (60,160 tokens), 10% for development test (7,520 tokens) and 10% for evaluation test sets (7,520 tokens). Overview of the division is in the Table 3.5.

Table 3.5.: Overview of corpus division

	Number of tokens	Number of entities
Training corpus	60,160	1332
Development corpus	7,520	186
Testing corpus	7,520	167

4. Theory and method

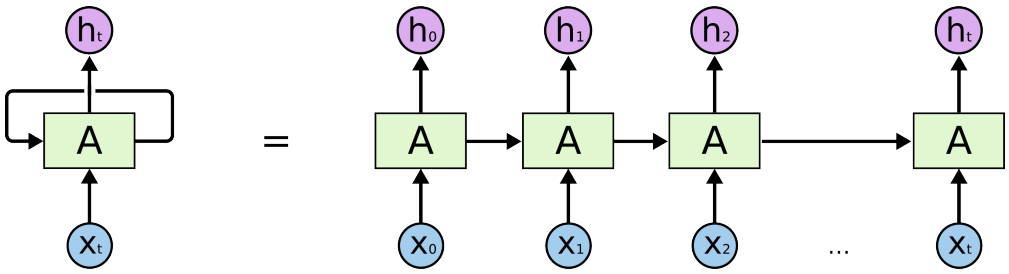

Figure 4.1.: An unrolled RNN
Source: http://colah.github.io/posts/2015-08-Understanding-LSTMs/

4.1. Neural Networks in general

Neural Networks are currently very popular in many fields including NLP. They are inspired by a model of the human neuron which are presented by small computing units. Each of these units takes a vector as an input and creates a single value as an output. Generally, we can divide neural networks into feed-forward networks (or multi-layer perceptrons) which do not use a cycle (loop) and reccurent neural networks (RNN) which include the cycles (loops). (Jurafsky and Martin, 2009)

4.2. Reccurent neural networks and Long Short-term Memory in general

RNNs are networks with loops which allows them to persist information. The graphical example can be seen in Figure 4.1., where A is a chunk of neural network, x_t is an input vector and h_t is an output.

However, RNN does not learn effectively long-term dependencies in a sequence (in our case in a text sequence). On the other hand, using an LSTM cell enable the RNN network to remember a piece of information for long periods of time. The chain of the LSTM cells is in usual RNN repeating moduls (signed as A in Figure 4.1.). Usual RNN has a simple structure of these moduls, they contains only one *tanh* layer. In contrast, the LSTM cell consists of four interacting layers.[1]

We can see a graphically expressed LSTM cell in Figure 4.2.. The input is composed of the current vector (x_t) and the previous output (h_{t-1}) in the figure. This input is then squashed by the *tanh* layer and passed by *input gate* which is

[1] http://colah.github.io/posts/2015-08-Understanding-LSTMs/

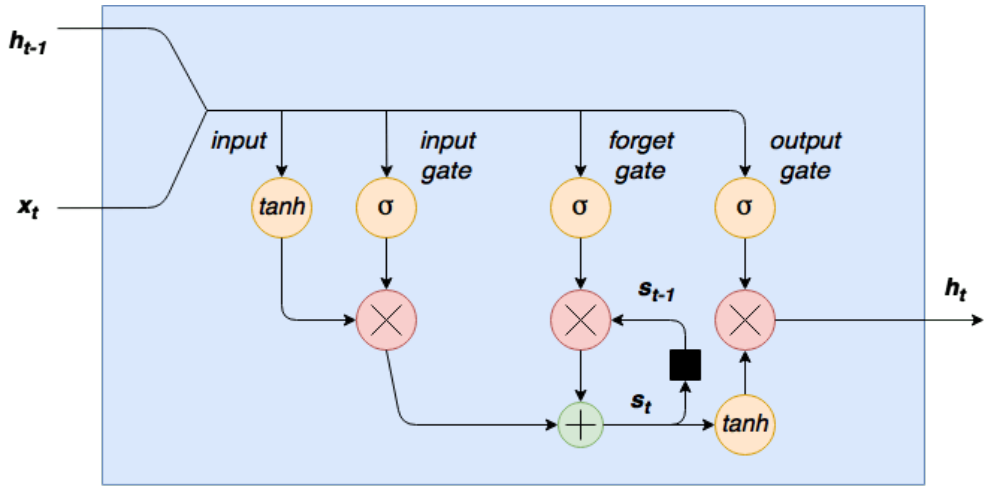

Figure 4.2.: LSTM cell diagram
Source: https://adventuresinmachinelearning.com/keras-lstm-tutorial

a sigmoid layer (σ). If the input vector elements are not required these sigmoid layers say how much these elements should be let through the gates (X). The output of the sigmoid function is a value between 0 and 1, respectively, where a value close to 0 means "switch off" and close to 1 means "let it through". Similarly, the *forget gate* controls if internal state variables (s_t) should be "remembered" or "forgotten". Moreover, the internal state variable creates an effective layer of recurrence as s_{t-1} is lagged one time step.[2]

Finally, the *output gate* consits of an output *tanh* function and another sigmoid function and the gate determines which values are allowed to be the final output of the cell (h_t).[3]

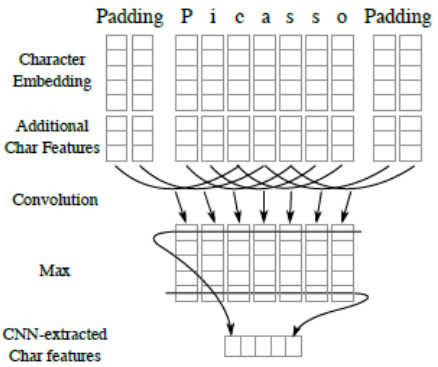

Figure 4.3.: Extraction of character features in the CNN
Source: Chiu and Nichols (2016)

Bidirectional RNN (bidirectional LSTM) means that the input data are fed into forward hidden layer and a backward hidden layer and they are connected

[2]https://adventuresinmachinelearning.com/keras-lstm-tutorial
[3]https://adventuresinmachinelearning.com/keras-lstm-tutorial

to get the same output. In such a case, the model considers information from backward as well as from forward at the same time.

4.3. Convolutional neural networks in general

Convolutional neural networks (CNN) are feed-forward neural networks which are usually used for image recognition and classification of objects. In our case, CNN is used for modeling character-level information in NER task. We can see an example of extraction of character features in Figure 4.3.. We can see there that the character embedding (and character type) feature vector are calculated based on a lookup table and then concatenated and passed into the CNN (Chiu and Nichols, 2016).

4.4. CNN-BiLSTM model

For our experiments, we created a hybrid bidirectional LSTM and CNN model based on work by Chiu and Nichols (2016) and implementation by Hofer (2018). We can see the model architecture in Figure 4.4. We decided for the model, because it achieved state-of-the-art results for English CoNLL-2003 data sets.

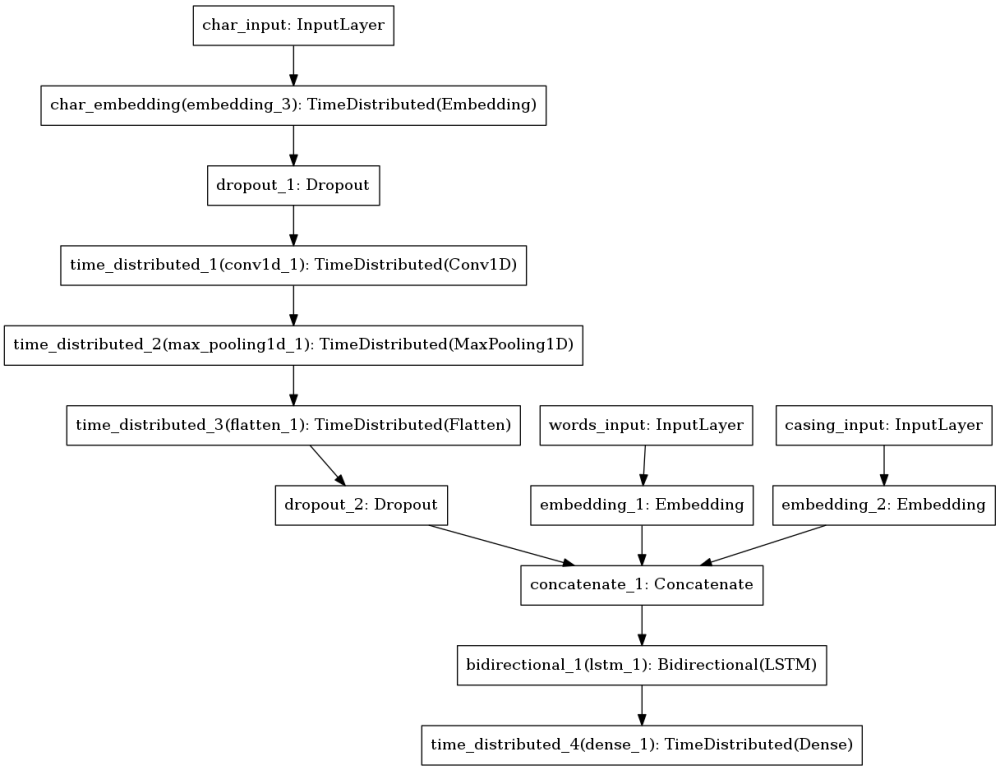

Figure 4.4.: Model architecture by Chiu and Nichols (2016)
Source: Hofer (2018)

The CNN-BiLSTM model combines bidirectional LSTM for sequence-labelling with CNN neural networks architecture that induce character-level features

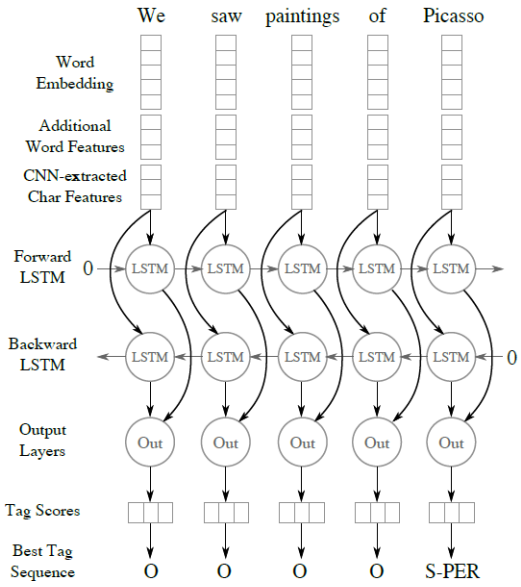

Figure 4.5.: Unrolled CNN-BiLSTM model
Source: Chiu and Nichols (2016)

(Figure 4.3). Therefore the model detects both character-level and word-level features. Moreover, this model has additional word-level input (casing input) that takes into consideration lower case and upper case of words (all lower, all upper, initial upper) as well as tokens which are numerals or includes numbers (numeric, mainly numeric, contains digit).

Basically, the model uses three embedding layers to map characters, words and casing to real numbers. All unique words are mapped to n-dimensional embedding vectors. These can be static computed by some semantic model[4] or randomly initialized and tuned during the training process. Then these vector representations are concatenated and fed into the bidirectional LSTM (BiLSTM) and the outputs of these two networks are added together. Finally, we use log-softmax activation layer to get a named entity tag score. We can see the unrolled CNN-BiLSTM model in Figure 4.5., and the output layers in detail in Figure 4.6..

Training of our model is provided on a per-sentence level and we used zero vectors as initial states for the LSTM. For training, we decided to apply adaptive learning rate optimizer Nadam (Nesterov-accelerated Adaptive Moment Estimation)[5], which performed the best result in preliminary experiments. However, we also tried to use mini-batch stochastic gradient descent (SGD) which does not improve results in comparison to Nadam. We also compared the results of Nadam and Adam (Adaptive Momement Estimation) optimizer that is a basic component of Nadam optimizer. In this case, the Nadam optimizer also showed slightly better results.

[4]Usually word2vec (https://code.google.com/archive/p/word2vec/) or fastText (https://fasttext.cc/) models are used.
[5]http://cs229.stanford.edu/proj2015/054_report.pdf

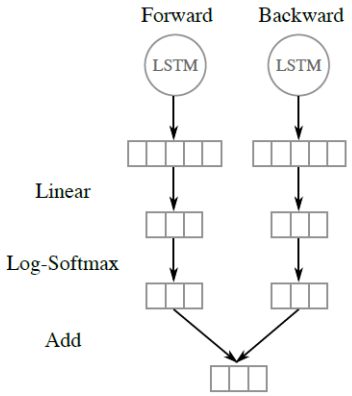

Figure 4.6.: The output layers in detail
Source: Chiu and Nichols (2016)

We implemented also dropouts to avoid overfitting. In Figure 4.4., we can see that the dropouts are imlemented after character embedding layer and before we concanated character embedded inputs to LSTM layer. In the case of LSTM layer, we applied dropout to output nodes.

Moreover, the CNN-BiLSTM by Chiu and Nichols (2016) used lexicons but for historical Czech, we do not have access to any suitable source for such a lexicon. Therefore our experiments do not work with them.

4.4.1. Embeddings

For our experiments, we used two different approaches to calculate the *word embeddings*. As the basic word embeddings, we used empty 50-dimensional vectors and we trained them during the experiments from their initial point.

We also tried to work with published pretrained word embeddings of contemporary Czech words provided by fastText[6]. These were trained on more than 178 millions of tokens from Wikipedia and 13 billions tokens based on common crawl (Grave et al., 2018).

The main reason for using these two approaches is the fact that there are no available pretrained word embeddings for historical Czech.

Similar to the model by Chiu and Nichols (2016), all the words were lowercased before they were passed to the lookup table to calculate their embeddings.

As we mentioned before, our model works with *character embeddings* as well. We used the same approach for embedding initialization as Chiu and Nichols (2016). It means that we randomly created a lookup table using values in range [-0.5,0.5] that were from uniform distribution to get 25-dimensional character embeddings. However, the set of characters include all the Czech characters (it means all characters with diacritical marks such as *á*, *ř*, *ť* or *ů*) and German vowels (e.g.: *ä*) because of German quotations in our datasets. We also include *padding* token (used in the CNN) and *unknown* token in the case of some other character appears.

[6] https://fasttext.cc/docs/en/crawl-vectors.html

Moreover, based on the implementation by Hofer (2018), we randomly initialized a separate lookuptable using 8 different capitalization features (casing input): *numeric*, *allCap*, *upperInitial*, *lowercase*, *other*, *mainly_numeric*, *contains_numbers* and *padding token*.

5. Evaluation and discussion

At first, we evaluated our system using standard metrics - *precision*, *recall* and *F1-score*. We compared the output of the NER system on the test set against its manually annotated gold standard version.

Moreover, we provided qualitative analysis for chosen linguistics phenomena to explore more deeply the problems of automatics recognition of historical NEs.

We decided to evaluate correct recognition of the basic NE type tag ("p", "i", "g", "t", "o" and "a") in the following experiments. It means that we did not distinguish "IOB" tags in the evaluation to see if NEs are tagged correctly no matter if they are at the beginning or inside (internal) of the NE (more about type of tags in Section 3.4.). This approach has allowed us to better show how chosen models work. However, NE type tags including the "IOB" tags should be evaluated in the future research within the project.

5.1. Comparison of different models

At first, we evaluated different combinations of model architectures to find out which model works the best for our data sets. We explored if the CNN model architecture (character-level input) and/or casing input improve F1 score for our test data set. We also experimented with pre-trained word embeddings by fastText for contemporary Czech (more in Section 4.4.1.) to explored if pretrained word embeddings should be considered in future research.

We used default hyper-parameters based on previous work by Chiu and Nichols (2016): *dropout* = 0.68; *recurrent dropout* = 0.25; *LSTM state size* = 275; *convolutional size* = 3; *learning rate* = 0.0105, *optimizer* Nadam and 20 *epochs*.

Table 5.1.: Precision, Recall and F1 score for different models. emb = fastText pretrained embeddings for contemporary Czech, cas = casing input (more in Section 4.4.)

Model	Precision	Recall	F1
LSTM	0.382	0.267	0.314
LSTM + cas	0.36	0.266	0.305
BiLSTM	0.438	0.277	0.34
BiLSTM + cas	0.477	0.282	0.354
BiLSTM + emb	0.582	0.321	**0.414**
BiLSTM + emb + cas	0.455	0.313	0.371
CNN-BiLSTM	0.473	0.300	0.367
CNN-BiLSTM + cas	0.507	0.27	0.355
CNN-BiLSTM + emb	0.483	0.326	0.389
CNN-BiLSTM + emb + cas	0.442	0.35	**0.39**

The evaluation (Table 5.1.) showed that the bidirectional LSTM achieved higher results than the one-directional LSTM itself. In the case of the BiLSTM model, the casing input also improve the F1 score. The experiments showed the similar improvement if we used fastText pre-trained word embeddings for the BiLSTM model in spite of the fact that these embeddings are trained on contemporary Czech language. However, the BiLSTM model using fastText word embeddings achieved better results if we did not implement the casing input layer.

In case of the CNN-BiLSTM model, we can see that adding casing input decreases the F1 score. Similar to the BiLSTM model, adding fastText pretrained word embeddings improve F1 score and there is almost no difference if we use casing input layer or not.

The fact that the casing input layer did not increase our F1-scores can be caused by OCR errors (especially random numbers which are not in original texts) in our data sets. The casing input takes into consideration if a token is numeric, mainly numeric or contains digits. The evaluation indicated that this approach does not affect the recognition of NEs in our historical data sets.

To sum up, using fastText pretrained word embeddings for contemporary Czech improved the results. As it was mention before in Section 4.4.1., these embeddings were trained on more than 175 millions of tokens. Therefore their embedding space is more structured (e.g. synonyms should be more likely to be represented by similar word vectors) in comparison to our other approach when we tuned randomly initialized vectors during the training process. Based on that, the pretrained embeddings help to correctly recognize NEs.

Moreover, if we compare the BiLSTM and the CNN-BiLSTM models (both using pretrained fastText embeddings), we can see that BiLSTM model reached higher F1 score (0.414) than CNN-BiLSTM (0.39). Based on that we could say that CNN did not improve our results neither. However, we assumed that hyper-parameter optimization could increase these results in case of CNN-BiLSTM model. Therefore we decided to optimize hyper-parameters for CNN-BiLSTM model in Section 5.2.

These results also indicated that training word embeddings specifically for historical Czech could improve our results and it could be one of the further experiments in the future.

5.2. Hyper-parameter optimization

To find the best setting for our model, we performed hyper-parameter optimization using the development set. Then we compared the output results of each settings (F1-score) to decide which combination has the most improving effect on the results. We used random search to select the best hyper-parameters. In Table 5.2., we can see the range of hyper-parameters from which we chose the settings and the final settings. All the optimization experiments in this section are based on the model with one of the best performance from the previous evaluation, it means the CNN-BiLSTM model using fastText pretrained embeddings and casing input layer. We also included hyper-parameters used by Chiu and Nichols

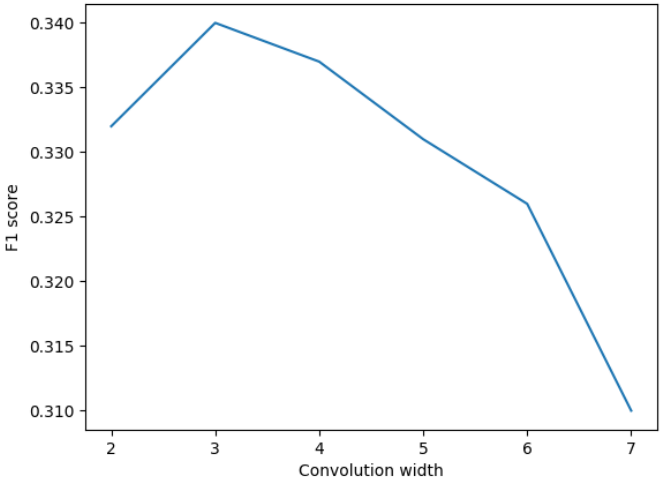

Figure 5.1.: Optimization of convolution width

(2016) for CoNLL-2003 data in the table to compare our final hyper-parameters with them.

We usually run chosen model five times with given value of a hyper-parameter and calculated average of F1-score. Final results of experiments with convolution width can be seen in Figure 5.1. We experimented with convolution width in range [2,7]. As we can see, the best convolution width is 3. This hyper-parameter achieved F1 score 0.34. In Figure 5.2., we can see results of experiments with a LSTM state size in range [100,500]. Based on these results, we can say that the best LSTM state size is 200 and larger size decrease results. Similarly, experiments with a learning rate in range [0.009,0.1] can be seen in Figure 5.3. The best results got same learning rate 0.01 as in default settings. We can see in the graph that larger learning rate decreased results. In Figure 5.4., we can see experiments with dropout values. We explored values in range [0.25, 0.95] and value 0.85 showed the best F1 score 0.393. Finally, we experimented also with number of LSTM layers in range [1,3] and these results can be seen in Table 5.3.. The best F1 score achieved model with 2 LSTM layers.

Table 5.2.: Overview of hyper-parameter optimization in comparison of settings by Chiu and Nichols (2016) for CoNLL-2003 data

Hyper-parameter	Our corpus		Chiu and Nichols (2016)
	Range	Final	CoNLL-2003
Convolution width	[2,7]	3	3
LSTM state size	[100,500]	200	275
LSTM layers	[1,3]	2	1
Learning rate	[0.01, 0.1]	0.01	0.0105
Epochs	-	7	80
Dropout	[0.25,0.95]	0.85	0.68

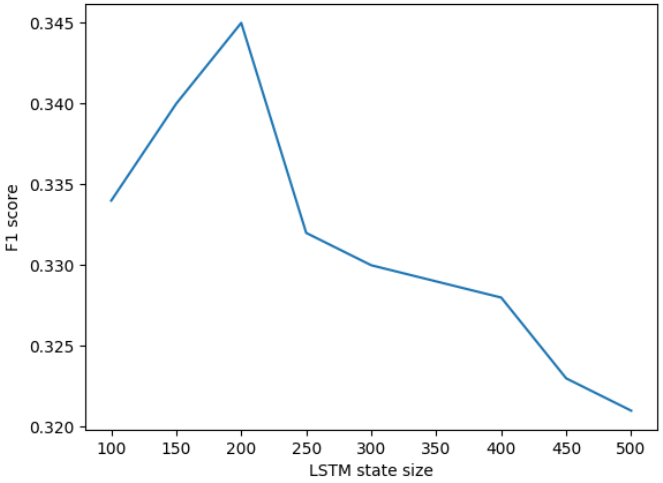

Figure 5.2.: Optimization of LSTM state size

Table 5.3.: Experiments with number of LSTM layers

Number of layers	F1 score
1	0.346
2	0.372
3	0.28

By the hyper-parameter optimization, we used LSTM state size 200 and dropout 0.85. We also decided to add one more BiLSTM layer.

Figure 5.2. shows the F1 score after every epoch. There it is shown that the model has the best performance around 7 epochs.

5.3. Final results

Based on the previous experiments with different combinations of model architectures and final hyper-parameter tuning, we evaluated our models again. The evaluation is provided based on the test data set. In Table 5.4., we can see the comparison of different models and their final precision, recall and F1 score. We also explored how our models recognize NEs in English CoNLL-2003[1] data sets to see how our models work for another language. We evaluated our models BiLSTM and CNN-BiLSTM with and without pretrained word embeddings. In case of our Czech data we used fastText word embeddings for contemporary Czech, for English, we used 50-dimensional Glove[2] word embeddings. However, in the case of English CoNLL-2003 data, we used different hyper-parameters based on experiments by Chiu and Nichols (2016): *dropout value* 0.68, *LSTM state size* 275, only one *LSTM layer* and 30 *epochs*.

[1] https://www.clips.uantwerpen.be/conll2003/ner/
[2] https://nlp.stanford.edu/projects/glove/

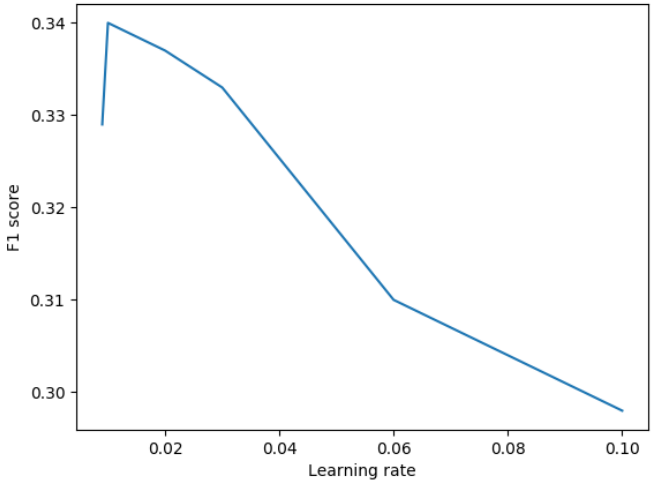

Figure 5.3.: Optimization of learning rate

Table 5.4.: Final results for chosen models using our historical Czech data and CoNLL-2003 data sets, emb = word embeddings (fastText for our corpus, Glove for English)

Model	Our corpus			CoNLL-2003 data		
	Prec	Rec	F1	Prec	Rec	F1
BiLSTM	0.426	0.289	0.345	0.704	0.561	0.624
BiLSTM + emb	0.56	0.343	0.425	0.799	0.872	0.803
CNN-BiLSTM	0.51	0.279	0.36	0.724	0.64	0.67
CNN-BiLSTM + emb	0.6	0.352	0.444	0.871	0.876	0.873

As we can see in Table 5.4., our models achieve higher F1 score for English CoNLL-2003 data sets in comparison to our corpus. The best results for English were reached by CNN-BiLSTM model using pretrained Glove embeddings, however, this model achieved only 0.444 F1 score for our Czech historical corpus. In our opinion, the main reason for such a difference is that the CoNLL-2003 data sets include more than 300,000 tokens in comparison to our 74,000. Moreover, the achieved numbers are for contemporary English using only NE categories *Location*, *Person*, *Organization* and *Miscellaneous*.

If we consider results from Czech NER state-of-art models (and ignore that we work with different data sets), we can say that we do not reach the best result in that field. Czech NER SVM-based model by Kravalova and Zabokrtsky (2009) reached F1 score 0.68 (more in Section 2.3.) and a model by Straková et al. (2013) based on Maximum Entropy Markov Model and a Viterbi algorithm achieved even 0.83 F1 score.

Similarly, in case of historical data, Grover et al. (2008) reached F1 score 0.72 for their rule-based NER system for English records of British parliamentary proceedings from the period 1814-1817, however, they evaluated only categories of personal names and names of places. However, we want to point out that both mentioned experiments used different data sets so we present them only for

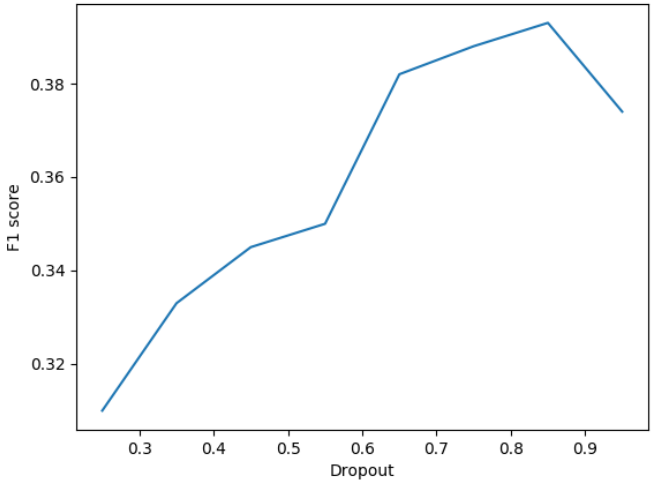

Figure 5.4.: Optimization of dropout values

illustration.

If we consider that there is no other NER research for Czech historical data, our results are promising to reach higher numbers in the future.

5.4. Qualitative analysis

For qualitative analysis, we compared tagged output text of the test data set to the same text that was manually annotated. We went through a half of the test data sets (3,725 tokens). The goal of the analysis was to observed in which cases the model automatically detects the NEs with correct tags and in which it does not. We also tried to describe the linguistic phenomena of historical Czech which cause problems for our CNN-BiLSTM model using pretrained fastText word embeddings for contemporary Czech.

The model performs satisfying output for the following NE types:

time expressions, especially in format *12. července 1872* ("12th July 1872") or abbreviated format *9. t. m.* (*9. tohoto měsíce*, "9th this month"),

geographical names, especially names of cities such as *Domažlice*, *v Trhanově* ("in Trhanov"), *z Petrohradu* ("from Petrohrad") or *do Solnohradu* ("to Solnohrad"). We included prepositions in the examples to show that they help the model to detect these NEs correctly (more below).

personal names, especially quite common Czech names or names common in input data sets, e.g. *Antonín*, *Jiří Prunař* or *Eduard Stoklas*,

artifact names, especially abbreviated names of currencies, e. g. *zl* (*zlatý*) and *kr* (*krejcar*) especially in the most common format *"number" zl. "number" kr.*

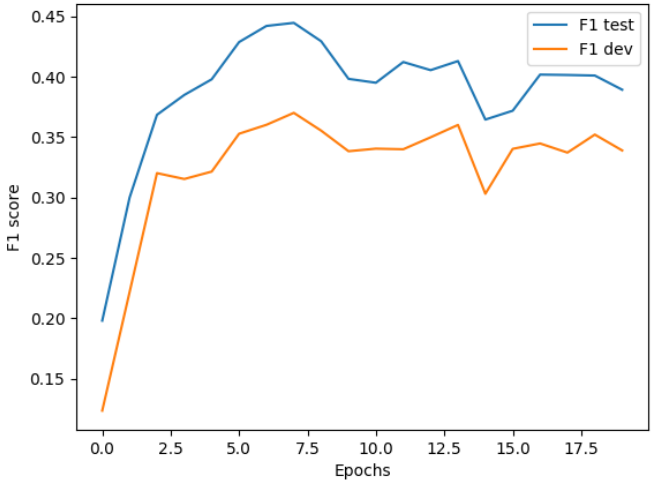

Figure 5.5.: Learning curve of CNN-BiLSTM model using pretrained word embeddings for contemporary Czech

In case of *institutions*, the automatic detection is promising, however, the tagging is almost never accurate, e. g., in NE *Akciový pivovar Staňkovský* ("Stock Brewery in Staňkov"), the first two tokens were tagged correctly, however, the token *Staňkovský* was not tagged at all. Similarly, tokens or pairs of tokens such as *nemocnice* ("hospital"), *pivovar* ("brewery") or *okres* ("district") and *zastupitelstvo obecní* ("municipal council") or *okresní hejtmanství* ("district office"), respectively, are tagged as NE type *institution* because they are often part of the institution names. These problems are caused by the fact that the names of institutions are usually multiwords expressions, and therefore harder to detect correctly.

Our model also had problems to correctly detect some of the other *geographical names*. Especially, if a previous token of the NE is an uncommon preposition in Czech. For example, in the word sequence *v Čechách, na Moravě a v Slezsku* ("in Bohemia, in Moravia, in Silesia"), the tokens *Čechách* and *Slezsku* were tagged correctly but the token *Moravě* was not tagged at all. Similarly, the sequence *ve Švihově u Klatov* ("in Švihov near Klatovy") where the token *Švihov* was tagged correctly and the token *Klatov* was not tagged, includes uncommon preposition *u*. Also, if a part of a multiword expression is common in other names, the model tags the part correctly, e.g. the token pair *Skleněná Huť*, the second one is more common and is tagged correctly, however, the first token is not.

Moreover, the model did not detect some *personal names* at all. In our opinion, these names are not common in the Czech language and therefore they are not in our input training dataset, e.g. *Christian Kotz*, *Wertzlera* (these names are more common in German) or *Paroubkovi* (however, very common Czech surname).

In case of *time expressions*, we found out that the model has problems to correctly detect expressions including names of days in the week or names of months in word forms which are not so frequent in the Czech language, e.g. *25. květnu* (more common word form *května*, "25th May") or *středu* (more common word form *středa*, "Wednesday") were not tagged.

Naturally, the model tagged some tokens which are not NEs, e.g. *nelibilo* ("disliked") or *okolí* ("surroundings"), some NEs were tagged with the wrong NE type, e.g. *Radnicích* (name of a town) or *Fridrichovi* ("Fridrich") or they were not tagged at all, e.g. *Jičíně* (name of town) or *Osvěta* (name of publication).

We would like to mention, that the analysis did not discover any wrong tagged NE because of its incorrect spelling. Based on that, we can say that spelling variety in historical texts is not an issue for our NN model.

Moreover, it is hard to say if the OCRed errors (especially dots and commas that are not in original texts) in our data sets caused any mis-tagged tokens. We have not find any wrong tagged token, however, we cannot rule out the possibility of that. Only problem caused by OCRed errors which we observed were sentences or tokens cut in the middle which could cause incorrect tagging.

To sum up, based on the analysis, we can say that described issues above are not caused by fact that the input data were historical texts and these issues could be partly solved by larger input data sets and therefore our results are promising for future research.

6. Conclusion and future work

We presented named-entity recognition for Czech historical texts using CNN-BiLSTM neural networks in the thesis. This work is a part of the Porta Fontium Project and the built system and obtained results will be used in further research within the project.

We decided to use neural networks methods to avoid a typical problem with historical data - the spelling difference in comparison to contemporary language. We can say that we successfully skipped the problem (more below).

However, our Czech historical data from the *Modern Access to Historical Sources Project* were obtained by OCRing the scanned originals and therefore we had to handle another issues. For example, the OCRed data contained errors and noise in form of characters and punctuation marks that were not in original texts. We tried to pre-process data by cleaning as much errors and noise as possible.

The most problematic OCR errors were random dots, exclamation marks and question marks which caused that some sentences or tokens were cut in the middle and could be tagged wrongly. Also, random numbers that were not in original texts probably caused that part of our model using casing input did not improve our final results (more in Section 5.1.).

For our data, we had to specify NE-types at first and we created annotation manual that should be used in further research activities within the project in the future. According to the manual, our data were manually annotated and divided into training, development and test data sets.

During our experiments, we evaluated different neural networks architecture (LSTM, BiLSTM, CNN-BiLSTM) and we compared their results to show which one is more effective in case of our historical data sets. We also experimented with two kinds of word embeddings - randomly initialized embeddings vectors trained during the training process and pretrained word embeddings for contemporary Czech by fastText.

We found out, that the CNN-BiLSTM model using pretrained fastText word embeddings has the best performance and achieves precision 0.6, recall 0.352 and F1 score 0.444. We also evaluated the model by using English CoNLL-2003 data and pretrained Glove embeddings and we reached 0.873 F1 score. Next to the other, these experiments showed that the pretrained word embeddings improved the obtained results, therefore it could be possible that pretrained word embeddings for Czech historical texts could improve results even more. However, to train own word embeddings could be problematic because we would need huge amount of historical data for that.

Another approach could be to use spelling normalization to get historical text close to the contemporary one and then try to use the word embeddings for contemporary Czech. However, this approach would lose the benefits why we decided to work with NN models.

We also provided qualitative analysis of observed linguistics phenomena. We found out that our CNN-BiLSTM model has problems to detect words, word forms and specific sequence of tokens that are uncommon in Czech language (and therefore they are not used enough in our training data set).

Based on these obtained results, we can say that neural networks are sufficient for detecting NEs in historical data. Moreover, the NN model is able to skip the problem with different spelling in historical texts in comparison to contemporary Czech. However, there is still problem of a variation in spelling (same words that are written differently in texts from the same period of time).

To sum up, we believe that if we prepare a larger training corpus of Czech historical texts that we carefully annotate using our annotation manual, we will achieve better results.

A. Annotation manual for NER in Czech historical texts

In the historical articles from the newspaper called *Posel od Čerchova* (1872) newspaper, we will annotate named entities (NE). The texts were first scanned and then converted to digitized text using OCR method (Optical Character Recognition).

We distinguish 6 basic tags:

- **p**: personal names
- **g**: geographical names
- **i**: institutions
- **t**: time expressions
- **o**: artifact names „objects“
- **a**: ambiguous (I believe it is a NE but I am not sure which one)

PERSONAL NAMES [p] include:

- first names and surnames (*Antonína, Kostlivýho, Antonína Kostlivýho, Julie M. Prushaková*, etc.),
- artistic names and nicknames (*Havlíček Borovský, Karel Hájek z Libočan, Mnich sázavský, Rudolf ze Stadionu, Jestřáb*, etc.),
- (academic) titels (*Med et Chir., Mistr, Dr., MUDr., Ing.*, etc.),
- royal (family) names, family names and names of historical persons (*Jan Lucemburský, Karel IV., Sámo, Lucemburkové, Habsburkové, Novákovi, Langobardi*, etc.),
- names of mythical and literary characters (*Přemysl Oráč, Švejk*, etc.).

INSTITUTIONS [i] include:

- names of institutions (*Československá obchodní akademie v Praze, Jenerální zastupitelství rakouského ústředního stavitelského spolku ve Vídni, C. k. vlastenecko-hospodářská společnost, občanská škola domažlická, Městská rada, Universita Odesská, Finanční výbor ve Vídni*, etc.),
- names of organizations and clubs (*Sbor ostrostřelecký, Stavitelský spolek ve Vídni, Sokol, Sokol domažlický, hasičský sbor Domažlice*, etc.),
- names of companies (*Cukrovar Domažlice, Tatra*, etc.),

- names of historical collectives, e.g. religious orders, political parties (*benediktini, husité, republikané*, etc.).

GEOGRAPHICAL NAMES [g] include:

- names of continents and (historical) states (*říše rakouská, Evropa, Čechy, habsburská monarchie, Rakousko-Uhersko*, etc.),
- names of (historical) territorial-administrative units (*panství Koutského a Trhanovského, Plzeňský kraj, okres podbořanský, Domažlicko, Bavory*, etc.),
- names of towns and their parts (*Pešť, Varšava, Plzeň, Horšův Týn, Nové Kdyně, Plzeň-Bory, Jižní Předměstí, Litice*, etc.),
- streets and public places (*poštovská ulice, dolejší předměstí v Domažlicích, Chodské náměstí, hradskou ulicí, Tomanova ulice*, etc.),
- names of natural monuments including local names (*vrch sv. Anny, dolech strousberských, Šumava, Mže, Úhlava, kopec Pohoří*, etc.),
- local names (*Svaté Dobrotivé, Na Hrázi, Pod Starým hradem*, etc.).

TIME EXPRESSIONS [t] include:

- names of date (*6. 2. 2019, 6. února 2019, 18. t . m.*, etc.),
- names of days (*středa, neděle*, etc.),
- names of hours (*12:00, v půl jedné*, etc.),
- names of months (*únor, listopad*, etc.),
- names of years (*MCCCLXXI, 1654, 1872*, etc.),
- names of centuries (*6. století př. n. l., 18. století, osmnácté století, 650 po Kristu*, etc.),
- names of epochs (*novověk, středověk, raný novověk, moderní doba, gotika, baroko*, etc.),
- holidays and important days (*Boží hod vánoční, Velikonoce, svátek Všech svatých, den sv. Josefa*, etc.),
- historic events (*bitvě na Bílé hoře, Pražská defenestrace, bitva u Slavkova*, etc.).

ARTIFACT NAMES ("objects") [o] include:

- names of documents (*Zlatá bula sicilská, Kutnohorský dekret*, etc.),
- names of artworks (*Hej Slované, opera Drahomíra, Vyšebrodský oltář, Krajinka v zimním hávu, Malá noční hudba*, etc.),
- names of products (*Turecké železniční losy, Uherské prémiové losy*, etc.),

- names of books and newspapers (*Posel od Čerchova, Svoboda, Osvěta, Životy posledních Rožmberků, Minulostí západočeského kraje, Pilsner Tagblatt*, etc.),
- names of buildings (*věž u svatých, kostel sv. Bartoloměje, zámek Kozel, klášter benediktinský u Davle*, etc.),
- names of currency (*kr, kr., zl, zl., zlatých, tolar,* etc.).

AMBIGUOUS [a]:
I can't distinguish. Anything I believe is an entity, but I am unable to determine which category it is.

Bibliography

Chiticariu, Laura, Yunyao Li, and Frederick R. Reiss (2013). "Rule-Based Information Extraction is Dead! Long Live Rule-Based Information Extraction Systems!" In: *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*. Seattle, Washington, USA: Association for Computational Linguistics, Oct. 2013, pp. 827–832. URL: https://www.aclweb.org/anthology/D13-1079.

Chiu, Jason and Eric Nichols (2016). "Named Entity Recognition with Bidirectional LSTM-CNNs". *Transactions of the Association for Computational Linguistics* 4, pp. 357–370. URL: http://aclweb.org/anthology/Q16-1026.

Collobert, Ronan, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa (2011). "Natural Language Processing (Almost) from Scratch". *J. Mach. Learn. Res.* 999888 (Nov. 2011), pp. 2493–2537. ISSN: 1532-4435. URL: http://dl.acm.org/citation.cfm?id=2078183.2078186.

Grave, Edouard, Piotr Bojanowski, Prakhar Gupta, Armand Joulin, and Tomas Mikolov (2018). "Learning Word Vectors for 157 Languages". In: *Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018)*.

Grover, Claire, Sharon Givon, Richard Tobin, and Julian Ball (2008). "Named Entity Recognition for Digitised Historical Texts". In: *LREC 2008*. URL: http://www.lrec-conf.org/proceedings/lrec2008/pdf/342_paper.pdf.

Hana, Jirka, Anna Feldman, and Katsiaryna Aharodnik (2011). "A Low-budget Tagger for Old Czech". In: *Proceedings of the 5th ACL-HLT Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pp. 10–18.

Hardmeier, Christian (2016). "A Neural Model for Part-of-Speech Tagging in Historical Texts". In: *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics*, pp. 922–931.

Hofer, Maximilian (2018). "Deep Learning for Named Entity Recognition #2: Implementing the state-of-the-art Bidirectional LSTM + CNN model for CoNLL 2003". Accessed: 2019-04-09.

Huang, Zhiheng, Wei Xu, and Kai Yu (2015). "Bidirectional LSTM-CRF Models for Sequence Tagging" (Aug. 2015). URL: http://arxiv.org/abs/1508.01991.

Jurafsky, Daniel and James H. Martin (2009). *Speech and Language Processing: An Introduction to Natural Language Processing, Computational Linguistics, and Speech Recognition*. Second. Pearson Prentice Hall.

Korchagina, Natalia (2017). "Normalizing Medieval German Texts: from rules to deep learning". In: *Proceedings of the NoDaLiDa 2017 Workshop on Processing Historical Language*, pp. 12–17.

Král, Pavel (2011). "Features for named entity recognition in Czech language". In: *In proceedings international conference on knowledge engineering and ontology development. Setúbal: SciTePress*, pp. 437–441.

Kravalova, Jana and Zdenek Zabokrtsky (2009). "Czech Named Entity Corpus and SVM-based Recognizer". In: *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration (NEWS 2009)*. Suntec, Singapore: Association for Computational Linguistics, pp. 194–201. URL: http://aclweb.org/anthology/W09-3538.

Lample, Guillaume, 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*. San Diego, California: Association for Computational Linguistics, pp. 260–270. DOI: 10.18653/v1/N16-1030. URL: http://aclweb.org/anthology/N16-1030.

Mac Kim, Sunghwan and Steve Cassidy (2015). "Finding names in Trove: named entity recognition for Australian historical newspapers". English. In: *Australasian Language Technology Association Workshop 2015*. Ed. by Ben Hachey and Kellie Webster. Vol. 13. Australasian Language Technology Association, pp. 57–65.

Neudecker, Clemens (2016). "An Open Corpus for Named Entity Recognition in Historic Newspapers". In: *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC 2016)*. Ed. by Nicoletta Calzolari (Conference Chair), Khalid Choukri, Thierry Declerck, Sara Goggi, Marko Grobelnik, Bente Maegaard, Joseph Mariani, Helene Mazo, Asuncion Moreno, Jan Odijk, and Stelios Piperidis. Portorož, Slovenia: European Language Resources Association (ELRA), May 2016. ISBN: 978-2-9517408-9-1.

Packer, Thomas L., Joshua F. Lutes, Aaron P. Stewart, David W. Embley, Eric K. Ringger, Kevin D. Seppi, and Lee S. Jensen (2010). "Extracting person names from diverse and noisy OCR text". In: *AND*. ACM, pp. 19–26.

Piotrowski, Michael (2012a). "Handling Spelling Variation". In: *Natural Language Processing for Historical Texts*, pp. 69–84.

Piotrowski, Michael (2012b). "Spelling in Historical Texts". In: *Natural Language Processing for Historical Texts*, pp. 11–24.

Rodriguez, Kepa J., Mike Bryant, Tobias Blanke, and Magdalena Luszczynska (2012). "Comparison of Named Entity Recognition tools for raw OCR text". In: Sept. 2012. DOI: 10.13140/2.1.2850.3045.

Ševcikova, Magda, Zdeněk Žabokrtský, and Oldřich Kruza (2007). "Zpracování pojmenovaných entit v českých textech". ÚFAL MFF UK.

Ševčíková, Magda, Zdeněk Žabokrtský, and Oldrich Kruza (2007). "Named Entities in Czech: Annotating Data and Developing NE Tagger". In: vol. Lecture Notes in Artificial Intelligence. Proceedings of the 10th International Conference Text, Speech and Dialogue (TSD 2007). Aug. 2007, pp. 188–195. DOI: 10.1007/978-3-540-74628-7_26.

Straková, Jana, Milan Straka, and Jan Hajic (2013). "A New State-of-The-Art Czech Named Entity Recognizer." In: *TSD*. Ed. by Ivan Habernal and Václav Matousek. Vol. 8082. Lecture Notes in Computer Science. Springer, pp. 68–75. URL: http://dblp.uni-trier.de/db/conf/tsd/tsd2013.html#StrakovaSH13.

Straková, Jana, Milan Straka, and Jan Hajič (2014). "Open-Source Tools for Morphology, Lemmatization, POS Tagging and Named Entity Recognition". In: Jan. 2014. DOI: 10.3115/v1/P14-5003.

Won, Miguel, Patricia Murrieta-Flores, and Bruno Martins (2018). "Ensemble Named Entity Recognition (NER): Evaluating NER Tools in the Identification of Place Names in Historical Corpora". *Frontiers in Digital Humanities* 5 (Mar. 2018). DOI: 10.3389/fdigh.2018.00002.