Detecting inflectional patterns for Croatian verb stems using class activation mapping

Domagoj Ševerdija[1], *Rebeka Čorić*[1], *Marko Orešković*[2], *and Lucian Šošić*[3]
[1] Department of Mathematics, University J. J. Strossmayer of Osijek
[2] National and University Library in Zagreb
[3] Faculty of Humanities and Social Sciences in Split

ABSTRACT

All verbal forms in the Croatian language can be derived from two basic forms: the infinitive and the present stems. In this paper, we present a neural computation model that takes a verb in an infinitive form and finds a mapping to a present form. The same model can be applied vice-versa, i.e. map a verb from its present form to its infinitive form. Knowing the present form of a given verb, one can deduce its inflections using grammatical rules. We experiment with our model on the Croatian language, which belongs to the Slavic group of languages. The model learns a classifier through these two classification tasks and uses class activation mapping to find characters in verbs contributing to classification. The model detects patterns that follow established grammatical rules for deriving the present stem form from the infinitive stem form and vice-versa. If mappings can be found between such slots, the rest of the slots can be deduced using a rule-based system.

Keywords: Croatian infinitive and present verb stems, convolutional neural network, class activation mapping

1 INTRODUCTION

Inflection of verbs in morphologically rich languages such as Croatian is important as it conveys grammatical information such as tense, aspect, mood, and voice, making sentences shorter and more precise. Computing the proper inflections of a verb is a well-established problem in computational morphology. Finding pairs of basic forms of Croatian verbs can convey enough information to apply the appropriate inflection rules. There is no way of setting the pairs right without applying brute force, i.e., checking each pair for each particular verb. Given the number of verbs in Croatian, checking manually can be time-consuming. Finding a mapping between basic verb forms can automate this process.

The choice of the proper verb form is in many ways arbitrary, as are most grammatical features. Many seemingly similar verbs belong to different conjugational paradigms for no apparent reason. Were it not so, creating a verb generator would be trivial. Instead, a sample of verbs can be used as input in order to obtain some statistical indicators on the likelihood of verbs with certain phonetical features appearing in certain conjugation classes, e.g.:

- Verbs ending in *-ati* that are derived from nouns or adjectives (*pilati, brzati*) have a strong tendency to take the present tense ending *-am*.
- Verbs ending in *-ati* that are loanwords (*krcati, peglati*) have a strong tendency to take the present tense ending *-am*. Searching for atypical phoneme clusters will thus be a statistical indicator of the ending.
- Verbs in *-ovati* take the present tense ending *-ujem*, except for some very short ones (*lovati→lovam*).

This view is by no means oversimplified; we do not suggest that the language is regular. There are irregularities such as *vreti*, which has an anomalous 3rd person plural. These anomalies in personal endings are very rare, however, and can be enumerated manually. The main goal of this paper is therefore to derive a model to find proper pairs of basic forms of Croatian verbs, conveying enough information to apply the appropriate inflection rules.

First, we shall describe our problem from a (broader) linguistic perspective and then give an overview of the current state-of-the-art computational approaches for tackling this problem. The main contribution is given in Section 2, with an example of rule-based system application in Section 3.4.

The Croatian verb system 1.1

The Croatian verb system is inherited from Proto-Slavic and conservatively preserves the ancient inflection. Consequently, the phonological and morphological rules governing conjugation are often opaque. There are six verb categories: **person**, **number**, **tense**, **mood**, **voice** and **aspect**. The first five categories are common in all Indo-European languages that preserve verbal inflection. Aspect can be either **perfective** or **imperfective**,[1] thus individual verbs are almost always either perfective or imperfective (a few can be biaspectual). Verb forms can be both finite and non-finite. Finite forms are conjugated in person and number. There are four moods: **indicative**, **imperative**, **optative**, and **conditional**, as described in Barić *et al.* 2005. The indicative mood has seven finite tenses: **present**, **perfect**, **aorist**, **imperfect**, **pluperfect**, **future**, and **perfective future**. Conditional has a present and a perfect form. This gives a total of 11 finite forms (Table 1).

	Present	Perfect	Aorist	Imperfect	Pluperfect	Future	Perfective Future
Indicative	●	●	●	●	●	●	●
Imperative	●						
Optative	●						
Conditional	●	●					

Table 1:
Finite verb forms

[1] Croatian aspects are referred to as "perfective" and "imperfective" in English. "Perfect" and "imperfect" are tenses. The traditional names are derived from similar roots, but are not the same. In fact, perfect tense can be both perfective and imperfective. Imperfect, however, is always imperfective.

Table 2: Thematic suffixes with final morpheme

Verb form	Stem	Thematic Suffix	Final morpheme
raditi 'to work'	*rad-*	*-i-*	*-ti*
radim 'I work'	*rad-*	*-i-*	*-m*
radih 'I worked'	*rad-*	*-i-*	*-h*
tresti 'to shake'	*tres-*	*-ø-*	*-ti*
tresem 'I shake'	*tres-*	*-e-*	*-m*
potresoh 'I shook'	*potres-*	*-o-*	*-h*

Non-finite verb forms (i.e. the **infinitive** and the **participles**) are not conjugated in person or number. Participles are not conjugated, but they can be declined as adjectives. They do not have standardized English names: here, they will be referred to as the l-participle, passive participle, present participle, and past participle. This gives a total of 5 non-finite forms. All of these forms can be synthetic or analytic. Synthetic forms are primary forms that consist of a single word. Analytic forms are derived by combining synthetic forms and auxiliary verbs. For example, the perfect tense is constructed using the l-participle and the auxiliary verb *biti* 'to be' in the present tense. The following forms are analytic: perfect, pluperfect, future, perfective future, optative, present conditional, and past conditional. The remaining 9 forms are synthetic: present, aorist, imperfect, imperative, l-participle, passive participle, present participle, past participle, and infinitive. Given that analytic forms can be derived from synthetic forms, describing the Croatian verb system can be reduced to deriving these 9 forms.

All synthetic forms are constructed from their base forms (stems). Stems are further modified with affixes (prefixes or suffixes) to produce different verb forms. Only suffixes are used to produce verbal conjugational forms. To describe the Croatian verb system, it is vital to properly parse the verbs – and identify which suffixes and stems exist. For an example of parsing, see the Table 2.

The purpose of Table 2 is to illustrate that stems can receive multiple morphemes, of two different types. Final suffixes, depend on the exact verb form (e.g. 1st person singular present). Each form has its characteristic suffix, uniform across the conjugations.[2] The thematic

[2] There are several irregularities in common verbs, e.g. 1st person singular *mogu* and *hoću*, and L-participle *išao*.

suffix, on the other hand, varies between different classes of verbs, as will be demonstrated in Section 1.2. The next task is to identify the stems. Nine synthetic verb forms referred to in Section 1.1 can all be considered to have their own stem, as discussed in Silić and Pranjković 2005. Such a situation is rather complex; fortunately, some of these stems are derived from others. For example, the present participle stem can be trivially derived from the present stem: it is always identical to 3rd person plural present tense followed by the suffix *-ći*. While other stems require more complex derivational rules, a regular derivation is still possible. In fact, all the stems can be regularly derived from two basic stems: the present and the infinitive. This is the "principal problem" of the "Slavic verbal system" (Micklesen 1974). The problem is referred to as "Slavic" since modern Slavic languages have all preserved old conjugational classes up to a point.

Historically, three types of verb classifications have been developed for the Slavic languages: infinitive stem, present stem, and basic stem classifications (Mihaljević 2014).

Infinitive stem classifications are the oldest, dating back to Dobrovskỳ (1809) who divided verbs according to the thematic suffix preceding the infinitive suffix *-ti*. To further describe the verbs, however, the present stem was required, so Dobrovský divided his first class into three groups (A, B, and C). This system was soon adapted for other Slavic languages. Present stem classifications prioritize present stems over infinitive stems when classifying verbs. However, the distinction is merely hierarchical, as both forms are still required to conjugate the verb properly.

Basic stem classifications were devised with an intent to derive the entire conjugation from a single stem. However, they still require knowing whether the basic stem is the present or the infinitive to work properly.[3]

Thus, given the knowledge of the present and the infinitive stems, the remaining Croatian verb forms can be derived regularly (minus the few irregularities in common verb forms, as stated above).

[3] For example, the classification of OCS (Old Church Slavic) verbs by Lunt (2001) differentiates between nine classes, each defined by a single classifier. The classifier is either the infinitive thematic suffix (for six classes) or the present suffix (for the remainder).

1.2 *Deriving the present from the infinitive and vice versa*

The first item which must be taken into account when deriving the present from the infinitive and vice versa is the **present thematic suffix**. As explained above, the thematic suffix is the suffix preceding the present final suffixes (*-m, -š, -ø, -mo, -te, -e/u*). The personal suffixes are always the same, so it is the thematic suffix that permits the existence of multiple present classes.

With regard to the **present ending (thematic suffix + personal suffix)**, up to five main groups are identified by grammarians: athematic, *-em, -im, -am,* and *-jem* present groups.

The *-em* and *-im* groups are fully regular. The athematic group can be considered irregular and, in standard Croatian, it consists of only the verb *biti* 'be'. The *-am* group is a contracted form of the *-em* group (e.g. *pěvajem→pjevam/I sing*), and the *-jem* group can be considered a subset of the *-em* group as well. However, *-am, -em,* and *-jem* verbs will be analyzed separately in this paper, in accordance with Croatian linguistic practice.

The other vital feature is the **infinitive thematic suffix**. It precedes the infinitive suffix *-ti*, and most grammarians isolate six infinitive thematic suffixes (*-ø, -nǫ, -ě, -i, -a* and *-ova*). The *-ø* suffix causes complex shifts, so verbs with various endings (*-eti, -sti, -rti, -ći*) shall be considered here. Besides the present and infinitive suffixes, there are further sound shifts (like ablaut) that render the conjugation less predictable. These will not be addressed herein, but the interested reader can consult Silić and Pranjković 2005.

1.3 *Previous work*

Over the past decade, the popularity of supervised methods has produced computational inflectional models for several morphologically-rich languages (see Durrett and DeNero 2013, Barros *et al.* 2017, and Dinu *et al.* 2012 and references therein).

There is a body of work that tries to give morphological transducers in the form of software components that performs morphological generation (e.g. for the Tulu language in Antony *et al.* 2012, the Hindi language in Goyal and Lehal 2008, the German language in Zielinski *et al.* 2009, Arabic languages in Habash and Rambow 2006 or for the Russian and the Ukrainian language in Korobov 2015). Most of the work uses some form of rule-based patterns which are defined by experts and are specific for each language. The algorithm then uses root words and appends suffixes based on these rules. In such an approach, each part of speech (e.g. nouns, verbs, or adjectives) has a different set of rules, which makes this approach very exhaustive.

For the Croatian language, there is a morphological generator actively being developed and used within the Croatian Online Syntactic and Semantic Framework described in Orešković *et al.* 2016.

In his PhD thesis, Wicentowski (2002) developed a minimally supervised framework of methods (combining supervised and unsupervised methods) for multilingual inflectional morphology covering 32 languages, but not including Croatian language, for the purposes of lemmatization. It is also worth noting that SIGMORPHON[4] (the Special Interest Group on Computational Morphology and Phonology) is a series of workshops and shared tasks focused on computational analysis of word structure in different languages, aiming to develop models that can generate word forms given linguistic information. It promotes research in computational morphology and phonology and in a recent shared task, SIGMORPHON 2023,[5] they asked the participants to design a model that learns to generate morphological inflections from a lemma and a set of morphosyntactic features of the target form for a broad range of languages. Each language in the task had its own training, development, and test datasets, but the Croatian was not provided. They also provided baselines (non-neural and neural models) for comparison.

Our contribution 1.4

The main contribution of the paper is a convolutional neural network model that takes Croatian verbs in infinitive stem form and classifies them into the appropriate present stem form and vice-versa. The classifier provides information that can be used by a transducer to

[4] `https://sigmorphon.github.io/`

[5] `https://github.com/sigmorphon/2023InflectionST`

compute the proper inflection of a given verb. Moreover, the model highlights feature maps that "voted" for proper classification, i.e. it highlights all the characters within a verb that were significant for classification. This is in line with the contemporary attempt to have "explainable" AI models (xAI) (for more information on xAI in NLP see Danilevsky *et al.* 2020). Compared to Wicentowski 2002, our model is relatively simple without any explicit feature design and supervision. On the other hand, the shared task "Part 1: Typologically Diverse Morphological (Re-)Inflection" from SIGMORPHON 2023 does provide a general framework of deriving inflections from a given lemma as an end-to-end system. In our setup, the model is used as an aid to the rule-based parser.

2 STEM MAPPING COMPUTATION

2.1 *The model*

We propose a neural-network-based computation model that learns to map Croatian verbs from the infinitive stem form to the present stem form and vice-versa, as described in Section 1.2. We refer to the former as INF2PRES and to the latter as PRES2INF problem. It is considered a classification problem. Our model is essentially a convolutional neural network and Section 3.3 empirically examines the appropriateness of such architecture.

As input, our model takes a single verb $x = \langle c_1, c_2, \ldots, c_n \rangle$ as a sequence of n characters c_i in infinitive and 1st person singular form respectively. Characters c_i are taken from a predefined alphabet V of bounded size and assigned a unique symbolic representation $c_i \in \{0,1\}^{|V|}$ (1-hot encoding). In our model, we use embedding $\mathbf{c}_i = \mathbf{E}c_i$ to compute dense vector representation of c_i, where $\mathbf{E} \in \mathbb{R}^{d_e \times |V|}$ and d_e is an embedding dimension. For a window of characters

$$\mathbf{x}_i = \mathbf{c}_{i:i+k_r-1} := \langle \mathbf{c}_i, \mathbf{c}_{i+1}, \ldots, \mathbf{c}_{i+k_r-1} \rangle$$

of size k_r, we apply a total of K l-channel 1D convolutions of size $k_r, r = 1, 2, \ldots, K$, which produces a feature vector:

$$\mathbf{f}_i^{(r)} = g(\mathbf{U}^{(r)}\mathbf{x}_i + \mathbf{b}^{(r)}) \in \mathbb{R}^l$$

for $i = 1, 2, \ldots, m_r$, where $\mathbf{U}^{(r)} \in \mathbb{R}^{l \times k_r d_e}$, $\mathbf{b}^{(r)} \in \mathbb{R}^l$. Therefore, $\mathbf{f}^{(r)}_{1:m_r}$ defines a feature map for an input x with respect to the r-th convolution. A wide convolution is applied: before the application of the filters, zero padding, if needed, is added before the first and after the last element of $\mathbf{x}_i$, making sure that the number of times that each character is included in the receptive field during convolution is the same, irrespective of the character's position in the word. Therefore, $m_r = n + k_r - 1$. A ReLU activation function is used for every convolutional layer: $g = \text{ReLU}$.

Every vector $\mathbf{f}^{(r)}_{1:m_r}$ obtained from a convolutional layer is max-pooled, which results in $K \times l$ 1-dimensional vectors. These vectors are concatenated into a new vector as $\mathbf{z} \in \mathbb{R}^{Kl}$, which is then relayed to a fully connected layer that outputs a score vector of dimension equal to the number of classes c:

(1) $$\mathbf{y}(x) = \mathbf{W}_{fc}\mathbf{z} + \mathbf{b}_{fc},$$

where $W_{fc} \in \mathbb{R}^{c \times Kl}, b_{fc} \in \mathbb{R}^c$. A softmax normalization is applied to vector $\mathbf{y}$ giving a probability vector over all classes

$$[P(x \in C_i)]_{i=1,2,\ldots,c} = \text{softmax}(\mathbf{y}(x)) \in [0,1]^c.$$

Before the linear layer, a dropout technique is used as a regularization method. The index of a maximum value in the resulting vector is the ordinal number of a class, namely, $C(x)$ to which verb x should be classified, thus:

$$C(x) = \underset{i \in \{1,2,\ldots,c\}}{\arg\max}\, P(x \in C_i).$$

See Figure 1 for an illustration. For the sake of simplicity, we denote our model as a function $C(x) = \text{CNN}_\Theta(x)$, where Θ are learned parameters for the model during training.

Class activation mapping 2.2

To obtain information about which characters contributed the most to the classification, class activation mapping (CAM) was used, as described in Lee *et al.* (2018). The main idea is as follows: for a given verb x, we compute a predicted class $C(x) = \text{CNN}_\Theta(x)$ and look for

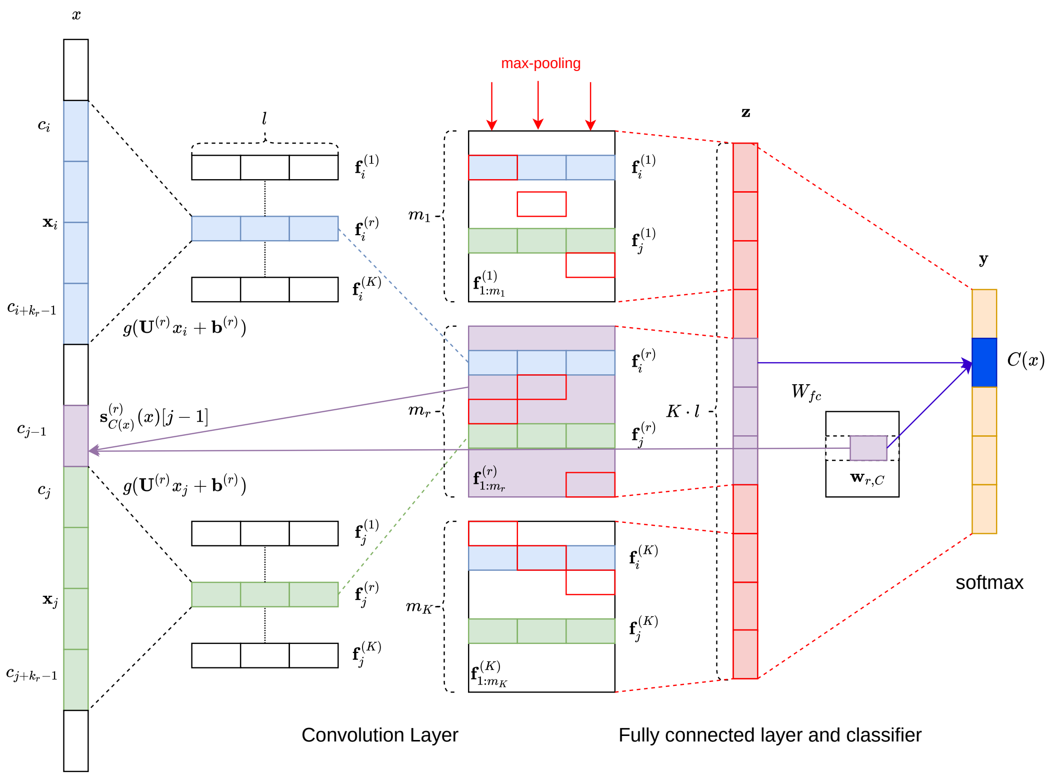


Figure 1: Two different windows of x, showing how they are used in convolution layers. The final output $C(x)$ is a predicted class for x and it is computed using all the available windows of x. The backward arrows indicate how the CAM filter was computed to score a contribution from c_{j-1} for the classification of x to class $C(x)$ with respect to the r-th convolution

characters of x whose feature maps are significant for the classification.

Let us rewrite (1) to consider the contribution of r-th convolution with l channels to the $C = i$ class (without loss of generality, we omit $\mathbf{b}_{fc}$ and assume $C \in \{1, 2, \ldots, c\}$):

$$\mathbf{y}(x)[i] = (\mathbf{W}_{fc}\mathbf{z})[i] = \sum_{r=1}^{K} \sum_{j=(r-1)l+1}^{rl} \mathbf{W}_{fc}[i,j]\mathbf{z}[j].$$

Note that $\mathbf{z}$ is a pooled vector by construction, we therefore want to apply the same weights on the entire feature map of r-th convolution, with $\mathbf{w}_{r,C} = W_{fc}[i,(r-1)l+1 : rl]$. Thus, we end up with a vector:

$$\mathbf{v}_C^{(r)} = \mathbf{f}_{1:m_r}^{(r)} \mathbf{w}_{r,C}^{\top}.$$

Unfortunately, $\mathbf{v}_C^{(r)}$ is a m_r-dimensional vector and depends on the type of convolution used, but it can be reduced to a fixed-size vector whose size is independent of the convolution. We achieve this using max-pooling with a window of size k_r and step 1 over $v_C^{(r)}$ deriving a vector:

$$\mathbf{s}_C^{(r)} = \left[\max\left(\mathbf{v}_C^{(r)}[p : p+k_r-1]\right)\right]_{p=1,2\ldots,n} \in \mathbb{R}^n.$$

The CAM returns a score for every character in x contributing to class C over all convolutions as:

$$\mathrm{CAM}(x,C) = \sum_{r=1}^{K} \mathbf{s}_C^{(r)}.$$

An illustration of CAM computation is shown in Figure 1. Examples of CAM application can be seen in Figure 4 (see page 57) and Figure 5 (see page 58).

EXPERIMENTS 3

Dataset 3.1

Our model was trained and evaluated on a set of Croatian verbs extracted from several lexical resources such as the Croatian WordNet (CroWN) described in Raffaelli *et al.* 2008, the Croatian linguistic portal (HJP),[6] and CroDerIV,[7] the Croatian lexicon of lexical and derivational morphemes by Šojat *et al.* (2012). The first resource was parsed using the NLTK[8] interface for the Open Multilingual Wordnet[9] by searching English verb synsets and retrieving lemmas in the Croatian language. These lemmas were used to query the CroDeriV and HJP

[6] http://hjp.znanje.hr
[7] http://croderiv.ffzg.hr/Croderiv
[8] https://www.nltk.org
[9] https://omwn.org/omw1.html

Table 3: Inf2Pres statistical overview

Classes	Train	Val	Test	Whole corpus
-am	36.04	35.96	35.95	36.02
-im	34.79	34.81	34.76	34.79
-jem	11.32	11.33	11.39	11.33
-em	17.85	17.90	17.90	17.86
Corpus breakdown	80.95	9.03	10.02	100.00

Table 4: Pres2Inf statistical overview

Classes	Train	VTest	Whole corpus	
-ati	53.82	53.42	53.23	53.73
-iti	32.12	31.76	31.82	32.05
-jeti	3.34	3.42	3.37	3.35
-eti	0.60	0.65	0.73	0.62
-uti	6.09	6.03	6.16	6.09
-sti	2.09	2.12	2.20	2.10
-rti	0.31	0.65	0.59	0.37
-ći	1.63	1.95	1.91	1.69
Corpus breakdown	80.80	9.10	10.10	100.00

search engines to obtain present stem forms. In addition, we queried HJP for verbs not found in CroWN and added them to the dataset.

The total number of verbs collected from these resources was 6794, manually organized as infinitive and present stem pairs for each verb. All pairs were verified by a human annotator.

The dataset for training and evaluation was organized as pairs (x_{inf}, x_{pres}), where x_{inf} denotes a verb in infinitive form and x_{pres} a 1st person singular present form. These forms are represented with appropriate suffixes as described in Section 1.2. The total number of available verbs is partitioned into train, validation, and test datasets with an 80:10:10 split by random sampling without replacement. A statistical overview of our dataset expressed as percentages is given in tabular form. Classes represent verb suffixes for the 1st person singular (Table 3) and infinitive form (Table 4). The most numerous classes are *-am* and *-im*, covering over 60% of the verbs for the Inf2Pres and *-ati* and *-iti* covering over 80% verbs in the Pres2Inf classification problem.

Results 3.2

The model was implemented in PyTorch ver. 1.8.0 (Paszke *et al.* 2019) and deployed on AMD Zen 12 CPU with 64GB of RAM and GeForce 2070 RTX GPU. The training time for both classification tasks took less than one minute per epoch and was trained for 30 epochs using the ADAM optimizer described in Kingma and Ba 2015. For initial character embeddings, we used FastText from Bojanowski *et al.* 2017, which gave slightly better results than random initialization.

Several hyperparameters of the CNN_{Θ} model had to be tuned before testing the model. These parameters were the number of filters, filter sizes, and dropout rate. Batch size and learning rate were also tuned for the training process. Hyperparameter tuning was conducted by exploring different values of parameters with 10-fold cross-validation. Parameters yielding the best average loss on validation sets were used to train the model. Hyperparameter tuning showed that, for both types of classifications, the same parameters can be used. The resulting parameters for the model can be seen in Table 5. The code is publicly available at a GitHub repository.[10]

Parameter	Value
l (number of filters)	36
filter sizes (k_r)	1, 2, 3, 5
dropout rate	0.1
batch size	50
learning rate	0.005

Table 5: Hyperparameters used for training the classifier

Table 6 reports the classification performance for our model on the test dataset for our model, in terms of accuracy and micro/macro/weighted F_1 scores. In both classification tasks, the model achieved relatively high scores in reported metrics. The quality of both classification tasks can be readily observed via confusion matrices given in Figure 2 and Figure 3. For example, the INF2PRES model classified 92% verbs that belong to the *-am* class accurately, and misclassified only 8%. In the PRES2INF model, the *-ati* verbs were

[10] `https://github.com/dseverdi/HR_verb_classification`

Table 6: Classification performance with and without FastText character embeddings

	Accuracy	Micro-F_1	Macro-F_1	Weighted-F_1
INF2PRES	0.896	0.896	0.877	0.896
+ FastText	**0.947**	**0.947**	**0.936**	**0.947**
PRES2INF	0.931	0.931	0.829	0.925
+ FastText	**0.947**	**0.947**	**0.834**	**0.943**

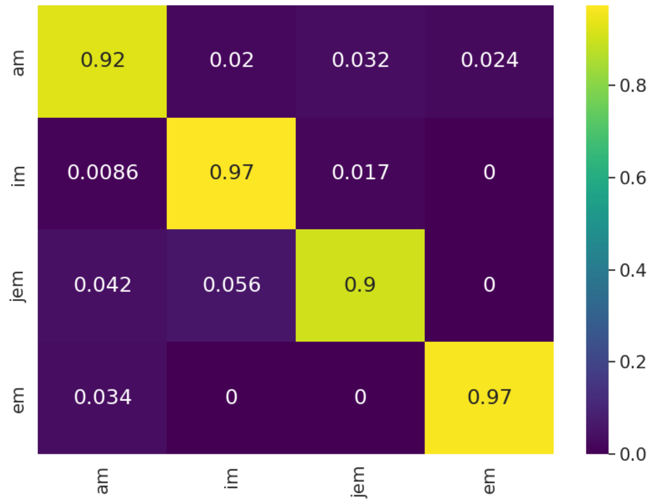


Figure 2: INF2PRES classification

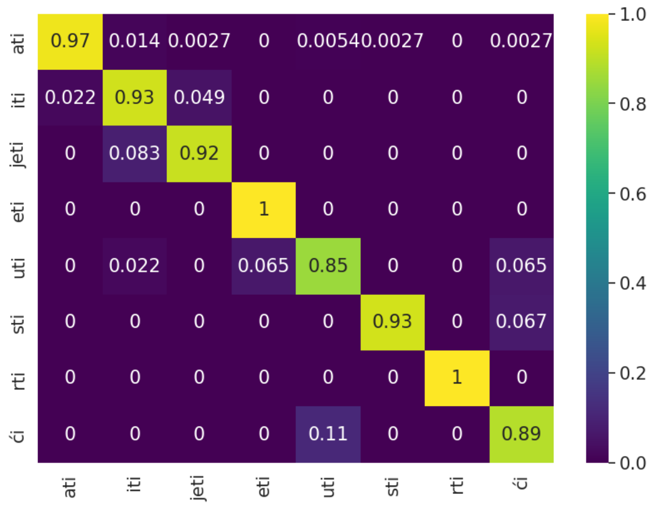


Figure 3: PRES2INF classification

classified properly in 97% cases, with only 3% of misclassifications. In a good classifier, diagonal values in confusion matrices should be as high as possible.

The interesting thing to see in our experiments are the CAMs for characters of verbs shown in Figure 4 and Figure 5.

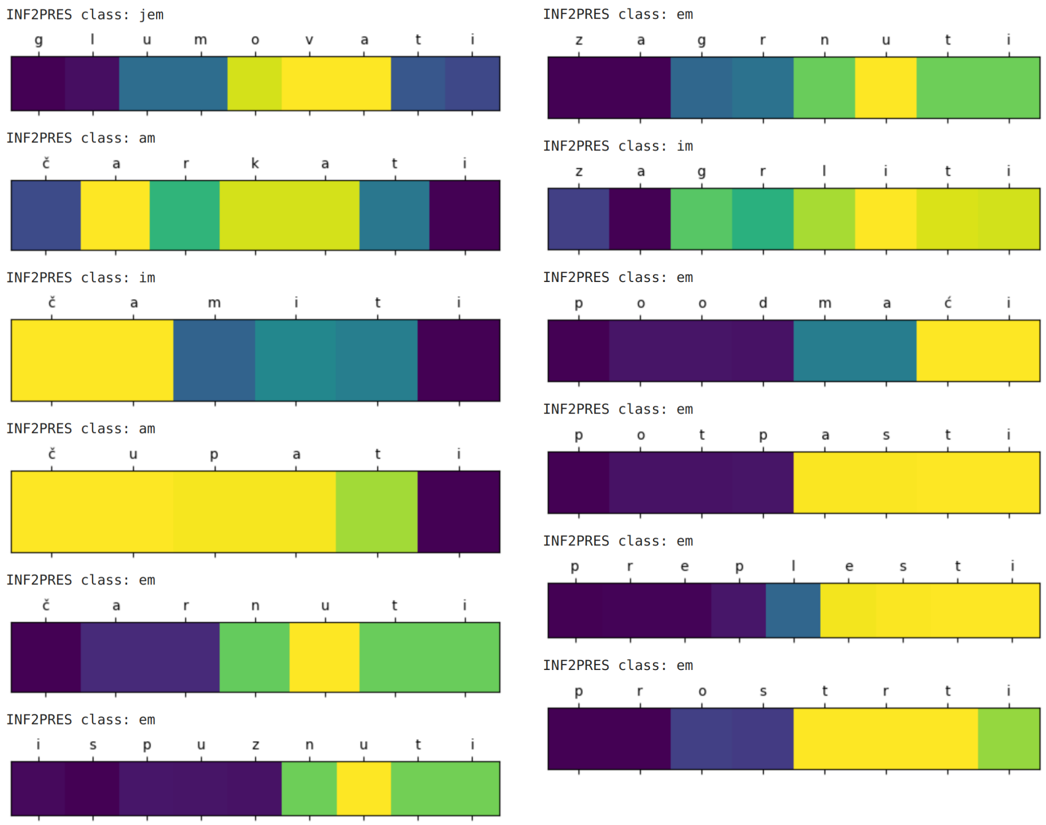


Figure 4: CAM filters for INF2PRES correctly predicted examples. Lighter colors indicate high contributing characters to classification

INF2PRES: Most CAM highlighted characters are usually suffixes of the verb with few exceptions. For example, verbs like *čarnuti* 'to ignite', *ispuznuti* 'to slide off', *zagrnuti* 'to cover', *zagrliti* 'to hold', *poodmaći* 'to go off', *potpasti* 'to fall under', and *prostrti* 'to lay down' follow this pattern. In some cases, infixes have more significance, as in the

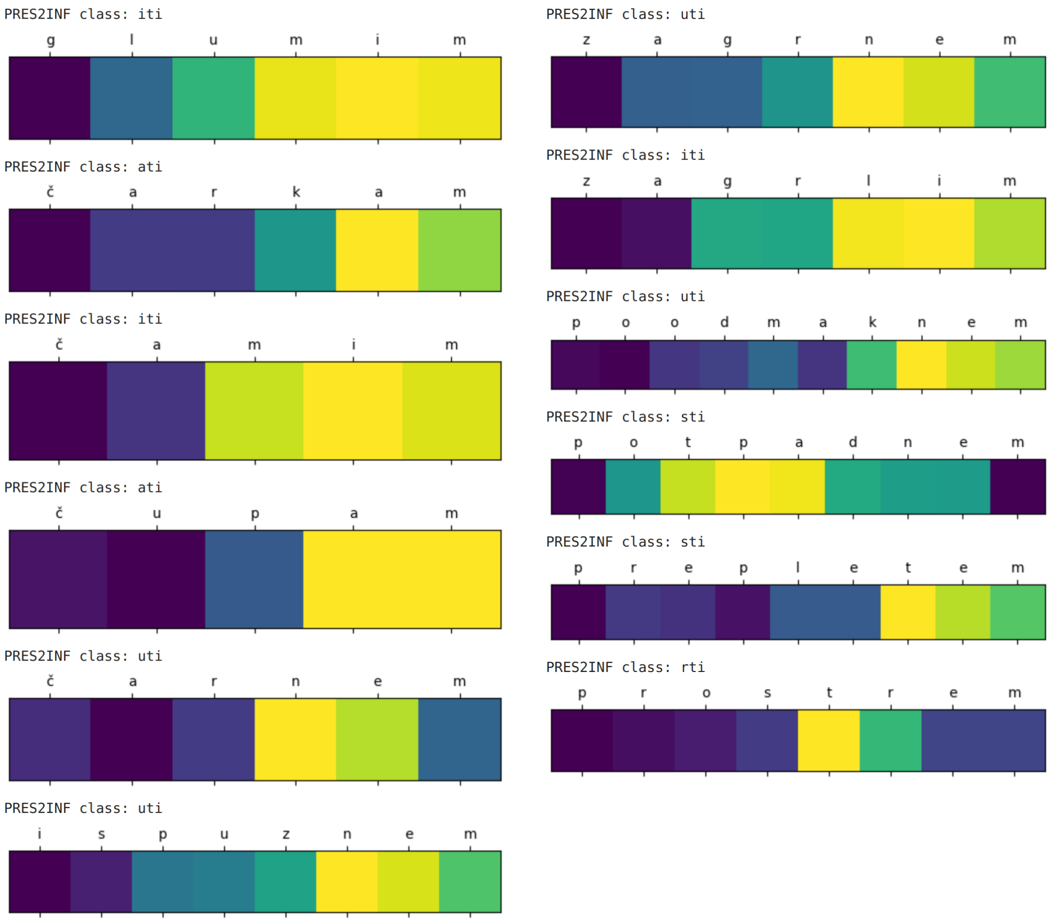


Figure 5: CAM filters for PRES2INF correctly predicted examples. Lighter colors indicate high contributing characters to classification

verb *glumovati* 'to act'. For the verbs *čarkati* 'to quarrel', *čamiti* 'to wait', and *čupati* 'to twitch', it highlights initial phoneme clusters more than suffixes, as the *-ati* class is more ambiguous.

PRES2INF: In most cases, the suffix has a significant role in classification. The significance of the infix is given in *potpadnem* 'I fall under' and *prostrem* 'I lay down'. For the latter, it is used to differentiate classes *-rti*, *-sti*. Note that, in both classification tasks, the model deals well with compound verbs (*odmaći*→*poodmaći*, *plesti*→*preplesti*, …).

Unsurprisingly, the CAM filters of Figures 4 and 5 show chaotic patterns of what the important letters contribute to classifying verbs into classes. Sometimes the important letters are at the beginning, sometimes in the middle, and sometimes at the end of the word. It seems that the classifier is having a hard time finding real regularities. However, this is somewhat expected. If it were not the case, such classification would be relatively straightforward. For example, the verbs *glumovati* 'to act' and *gladovati* 'to starve' have the first person present form *glumujem* and *gladujem*, respectively. In contrast, the verb *glumatati* 'to pretend' has the first person present *glumim*. The model in this example is not prone to give importance to suffixes because, as a feature, they are not significant for classification. However, if the model attends more to the morphemes -*ova*- and -*at*- while considering that both verbs end with -*ati*, it can infer what the appropriate classes are.

Ablation study 3.3

In this section, we consider the importance of specific architectural concepts of our model, namely:

- *importance of windowing*: The baseline model denoted as FF^{w} is a feed-foward neural network with with two layers. The top layer is a softmax classifier. The purpose of the first layer is to find a mapping of aggregated information from the characters within a window of size w (concatenation of character vectors within a window). The second layer aims to find high-level features for the classification.
- *importance of convolutions*: a CNN model denoted CNN$^{\{k_r\}}$ described in Section 2.1, with the list of 1D convolution sizes $\{k_r\}$ with a total of l channels.

All models use ReLU as an activation function and are trained using cross-entropy loss using the ADAM optimizer. The metaparameters are set as in Table 5 and FastText pretrained character vectors are used. No model performance degradation due to class imbalance was observed using standard cross-entropy training compared to cross-entropy with class weights.

Table 7: INF2PRES models performance. Arrows indicate whether greater or lower is better

Model	Accuracy ↑	RMSE ↓	Micro-F1 ↑	Macro-F1 ↑	Weighted-F1 ↑
FF^1	0.859	0.881	0.835	0.859	0.859
FF^5	0.885	0.781	0.863	0.885	0.884
$CNN^{\{5\}}$	0.939	0.565	0.939	0.928	0.939
$CNN^{\{1,2,3,5\}}$	0.941	0.534	0.941	0.927	0.947

Table 8: PRES2INF models performance. Arrows indicate whether greater or lower is better

Model	Accuracy ↑	RMSE ↓	Micro-F1 ↑	Macro-F1 ↑	Weighted-F1 ↑
FF^1	0.894	1.016	0.643	0.894	0.879
FF^5	0.913	0.773	0.703	0.913	0.907
$CNN^{\{5\}}$	0.946	0.529	0.946	0.783	0.943
$CNN^{\{1,2,3,5\}}$	0.950	0.467	0.950	0.835	0.948

In both Table 7 and Table 8, one can observe that the addition of windows improves performance over the baseline model. The reason for this is that windows make it possible to capture the local context of characters. Moreover, the filtering of characters with only one convolution with l filters was beneficial for the model. We believe that multiple channels in convolution enabled the capture of several aspects of features for classification. The addition of several convolution sizes slightly improved the overall result.

3.4 *Experiments with SSF*

The SSF (Orešković *et al.* 2016)[11] contains a rule-based morphological generator (MG) for expanding its Croatian lexicon. It is written in Python and included in the SSF as a web service. The whole of SSF's lexicon was processed initially by the MG, manually corrected and published as an online resource in the Linguistic Linked Open Data cloud (Orešković *et al.* 2018). The MG in general takes a lemma of

[11] SSF is publicly available at `http://ss-framework.com/?lang=en`.

	Accuracy	RMSE	Micro-F_1	Macro-F_1	Weighted-F_1
SSF INF2PRES	0.667	0.047	0.667	0.635	0.658

Table 9: SSF rule-based morphological parser performance on test set

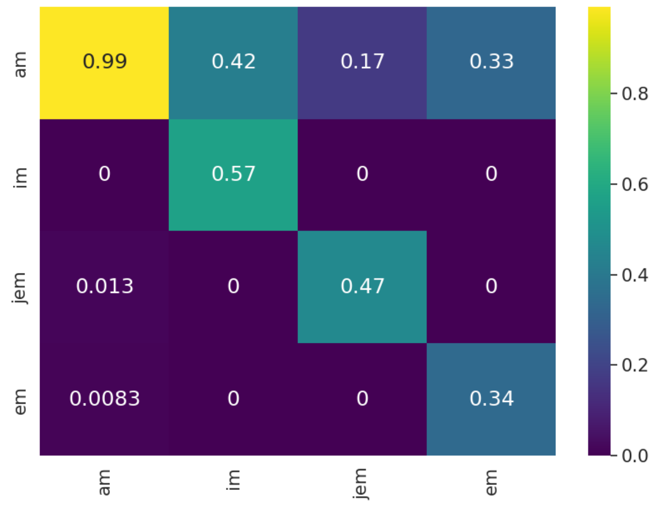
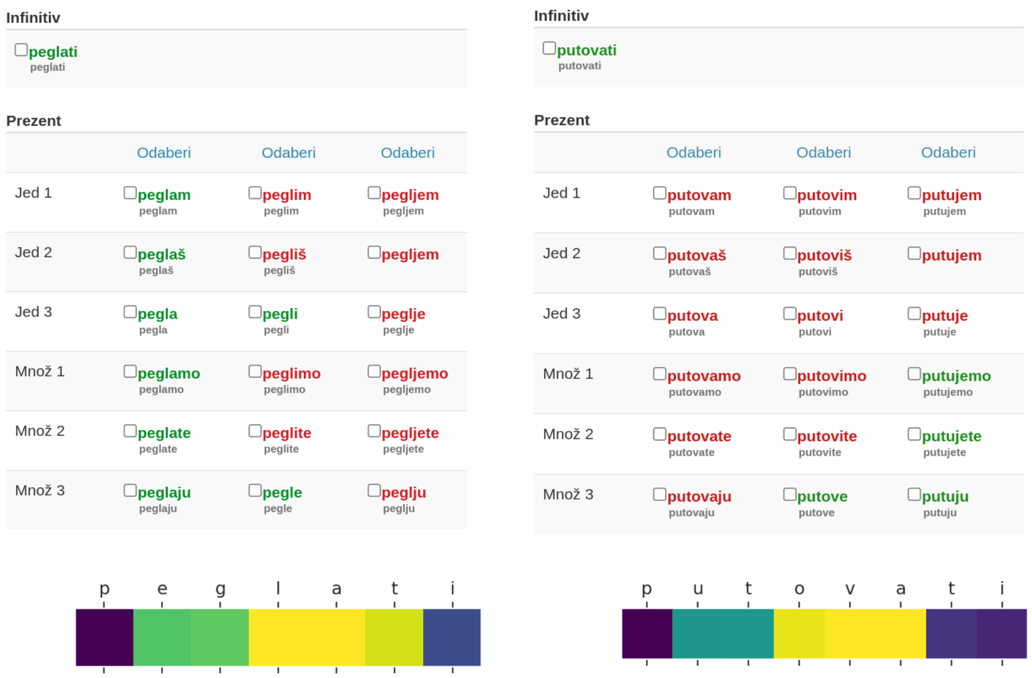


Figure 6: SSF INF2PRES classification

the word and tries to find its proper inflections by applying specific grammatical rules. In the current state, it does not use any statistical information about words. Specifically for verbs, it takes an infinitive as an input and applies cascading rules that extract the root of the verb by subtracting known suffixes. Once the root is extracted, MG merges root and suffix for each verb form. For present tense, suffixes are: *-em*, *-im*, *-jem*, *-am*. After the root and suffix are merged, the MG applies sound changes to that newly formed word (e.g. sibilarization, palatalization, iotation, etc.). Using a strictly rule-based approach, MG ends up with several equally probable paradigms (i.e. it applies at least one of the possible present tense suffixes). It is worth noting that it is still in development and using CAMs from INF2PRES model, and it can help us derive, to a certain extent, meaningful rules for better MG transduction. The current performance of the SSF for INF2PRES classification is given in Table 9 and Figure 6, if we choose only one paradigm (i.e. the first one). We do not apply it on PRES2INF because it is primarily designed for infinitive input.



Infinitiv

☐peglati peglati

Prezent

	Odaberi	Odaberi	Odaberi
Jed 1	☐peglam peglam	☐peglim peglim	☐pegljem pegljem
Jed 2	☐peglaš peglaš	☐pegliš pegliš	☐pegljem
Jed 3	☐pegla pegla	☐pegli pegli	☐peglje peglje
Množ 1	☐peglamo peglamo	☐peglimo peglimo	☐pegljemo pegljemo
Množ 2	☐peglate peglate	☐peglite peglite	☐pegljete pegljete
Množ 3	☐peglaju peglaju	☐pegle pegle	☐peglju peglju

Infinitiv

☐putovati putovati

Prezent

	Odaberi	Odaberi	Odaberi
Jed 1	☐putovam putovam	☐putovim putovim	☐putujem putujem
Jed 2	☐putovaš putovaš	☐putoviš putoviš	☐putujem
Jed 3	☐putova putova	☐putovi putovi	☐putuje putuje
Množ 1	☐putovamo putovamo	☐putovimo putovimo	☐putujemo putujemo
Množ 2	☐putovate putovate	☐putovite putovite	☐putujete putujete
Množ 3	☐putovaju putovaju	☐putove putove	☐putuju putuju

Figure 7: This snippet of MG conjugation shows 3 possible inflections for the verbs *peglati* and *putovati* (only present is shown). Our classifier predicts which inflection is suitable: *peglati*→*peglam*, *putovati*→*putujem*. Note that both examples have the same suffix -*ati* but have different present stems

For example (see Figure 7), MG yields 3 groups for the verb *peglati* ‘to iron’ and our model picks the correct one:

- *pegl* + *am*
- **pegl* + *im*
- **pegl* + *jem*

CAM: attend to the stem boundary and thematic suffix -*a* and start of the final morpheme.

For the verb *putovati* ‘to travel’, MG produces also 3 possible conjugations (with -*ova* as a thematic suffix) and our model picks the correct one:

- **put* + *ov* + *am*
- **put* + *ov* + *im*
- *put* + *u* + *jem*

CAM: attend more to the thematic suffix *-ova* and start of the final morpheme.

In these examples, CAM shows the significance that the model assigns to each character with regard to the choice of present final morphemes. For future work, this can be helpful to define rules for MG to capture the proper inflection.

Experiments with SIGMORPHON 2023 baselines 3.5

We compare our results with the baselines of SIGMORPHON 2023 Task 0 (Part 1), namely:

- `non-neural model`: a simple model that tries to align input/output examples during the training using Levenshtein distance and deduce appropriate prefix and suffix changing rules for given examples
- `neural model`: a Transformer based model applied for character level transduction from Wu *et al.* 2021.

Both models are implemented and publicly available.[12]

In our setup, they were trained and validated on INF2PRES datasets as generative models, i.e., they predict the proper inflected verb for the given infinitive. We treat the problem as a classification task.

Table 10 shows results for transducing Croatian verbs from infinitive (lemma) to first person present. For comparison, we also show our CNN model combined with MG for INF2PRES transduction. Although SIGMORHPON models achieve relatively worse results in our setup, they should be the first choice if datasets are large enough so that these models can learn general inflections (not constrained to verbs only). Our approach is more restricted, and it is useful if data is scarce and if rule-based systems are available (which is the case for the Croatian language).

[12] `https://github.com/sigmorphon/2023InflectionST/tree/main/part1/baselines`

Table 10:
SIGMORPHON 2023 baselines
for Croatian verb transduction

Model	Accuracy
non-neural	0.8786
neural	0.8994
CNN + MG	0.9470

4 CONCLUSION

In this paper, we provide an overview and motivation for the Croatian verb classification problem as a particular case of the Slavic inflection system. A neural network model with class activation mapping was applied as a supervised learning model on collected datasets. It is the initial step in applying present and infinitive stem classifiers in conjugating Croatian verbs. From this point on, one can apply rule-based transducers designed explicitly for the Croatian language (SSF by Orešković *et al.* 2018 would be one example) or apply some of the tools available on the market. If there is an abundance of data, one should resort to the established state-of-the-art models available via SIGMORPHON shared tasks.

Following the recent trends in natural language processing, the shift from rule-based and predictive models (supervised learning) to generative or unsupervised models becomes an interesting approach in inflectional morphology, especially for morphologically rich languages like Croatian. There are some promising results that encourage this pursuit, such as those in Şulea and Young 2019.

REFERENCES

P. J. ANTONY, Hemant B. RAJ, B. S. SAHANA, Dimple Sonal ALVARES, and Aishwarya RAJ (2012), Morphological analyzer and generator for Tulu language: A novel approach, in Sabu M. THAMPI, El-Sayed EL-AFRY, and Javier AGUIAR, editors, *Proceedings of the International Conference on Advances in Computing, Communications and Informatics*, ICACCI '12, pp. 828–834, Association for Computing Machinery, New York, NY, USA, doi:10.1145/2345396.2345531.

Eugenija BARIĆ, Mijo LONČARIĆ, Dragica MALIĆ, Slavko PAVEŠIĆ, Mirko PETI, Vesna ZEČEVIĆ, and Maja ZNIKA (2005), *Hrvatska gramatika,* Školska knjiga, Zagreb, ISBN 9789530400108.

Cristina BARROS, Dimitra GKATZIA, and Elena LLORET (2017), Inflection generation for Spanish verbs using supervised learning, in Manaal FARUQUI, Hinrich SCHUETZE, Isabel TRANCOSO, and Yadollah YAGHOOBZADEH, editors, *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pp. 136–141, Association for Computational Linguistics, Copenhagen, Denmark, doi:10.18653/v1/W17-4120.

Piotr BOJANOWSKI, Edouard GRAVE, Armand JOULIN, and Tomas MIKOLOV (2017), Enriching word vectors with subword information, *Transactions of the Association for Computational Linguistics*, pp. 135–146, doi:10.1162/tacl_a_00051.

Marina DANILEVSKY, Kun QIAN, Ranit AHARONOV, Yannis KATSIS, Ban KAWAS, and Prithviraj SEN (2020), A survey of the state of explainable AI for natural language processing, in Kam-Fai WONG, Kevin KNIGHT, and Hua WU, editors, *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pp. 447–459, Association for Computational Linguistics, Suzhou, China.

Liviu P. DINU, Vlad NICULAE, and Octavia-Maria SULEA (2012), Learning how to conjugate the Romanian verb. Rules for regular and partially irregular verbs, in Walter DAELEMANS, editor, *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 524–528, Association for Computational Linguistics, Avignon, France.

Josef DOBROVSKỲ (1809), *Ausführliches Lehrgebäude der Böhmischen Sprache, zur gründlichen Erlernung derselben für Deutsche, zur vollkommenern Kenntniß für Böhmen*, J. Herrl, Prague, `https://books.google.fr/books?vid=UOM:39015036760190&redir_esc=y`.

Greg DURRETT and John DENERO (2013), Supervised learning of complete morphological paradigms, in Lucy VANDERWENDE, Hal DAUMÉ III, and Katrin KIRCHHOFF, editors, *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1185–1195, Association for Computational Linguistics, Atlanta, Georgia, `https://aclanthology.org/N13-1138`.

Vishal GOYAL and Gurpreet Singh LEHAL (2008), Hindi morphological analyzer and generator, in Preeti R. BAJAJ, Amol Y. DESHMUKH, and Kailash D. JOSHI, editors, *2008 First International Conference on Emerging Trends in Engineering and Technology*, pp. 1156–1159, doi:10.1109/ICETET.2008.11.

Nizar HABASH and Owen RAMBOW (2006), MAGEAD: A morphological analyzer and generator for the Arabic dialects, in Nicoletta CALZOLARI, Claire

CARDIE, and Pierre ISABELLE, editors, *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pp. 681–688, Association for Computational Linguistics, Sydney, Australia, doi:10.3115/1220175.1220261.

Diederik P. KINGMA and Jimmy BA (2015), ADAM: A method for stochastic optimization, in Yoshua BENGIO and Yann LECUN, editors, *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7–9, 2015, Conference Track Proceedings*, doi:10.48550/arXiv.1412.6980.

Mikhail KOROBOV (2015), Morphological analyzer and generator for Russian and Ukrainian languages, in Mikhail Yu. KHACHAY, Natalia KONSTANTINOVA, Alexander PANCHENKO, Dmitry IGNATOV, and Valeri G. LABUNETS, editors, *Analysis of Images, Social Networks and Texts*, pp. 320–332, Springer International Publishing, Cham, doi:10.1007/978-3-319-26123-2_31.

Gichang LEE, Jaeyun JEONG, Seungwan SEO, CzangYeob KIM, and Pilsung KANG (2018), Sentiment classification with word localization based on weakly supervised learning with a convolutional neural network, *Knowledge-Based Systems*, 152(C):70–82, ISSN 0950-7051, doi:10.1016/j.knosys.2018.04.006.

Horace G. LUNT (2001), *Old Church Slavonic grammar*, Mouton de Gruyter, The Hague, doi:10.1515/9783110876888.

Lew R. MICKLESEN (1974), The common Slavic verbal system, in Ladislav MATEJKA, Victor TERRAS, and Anna CIENCALA, editors, *Vol. 1 Linguistics and Poetics*, chapter American contributions to the Seventh International Congress of Slavists, August 21–27, 1973, pp. 241–274, De Gruyter Mouton, Berlin, Boston, ISBN 9783110873948, doi:10.1515/9783110873948-011.

Milan MIHALJEVIĆ (2014), *Slavenska poredbena gramatika 2. dio: Morfologija, prozodija, slavenska pradomovina.*, Školska knjiga, Zagreb, ISBN 953-0-30225-8.

Marko OREŠKOVIĆ, Sandra LOVRENČIĆ, and Mario ESSERT (2018), Croatian Network Lexicon within the Syntactic and Semantic Framework and LLOD Cloud, *International Journal of Lexicography*, 32(2):207–227, ISSN 0950-3846, doi:10.1093/ijl/ecy024.

Marko OREŠKOVIĆ, Jakov TOPIĆ, and Mario ESSERT (2016), Croatian linguistic system modules overview, in George Meladze TINATIN MARGALITADZE, editor, *Proceedings of the 17th EURALEX International Congress*, pp. 280–283, Ivane Javakhishvili Tbilisi University Press, Tbilisi, Georgia, ISBN 978-9941-13-542-2.

Adam PASZKE, Sam GROSS, Francisco MASSA, Adam LERER, James BRADBURY, Gregory CHANAN, Trevor KILLEEN, Zeming LIN, Natalia GIMELSHEIN, Luca ANTIGA, Alban DESMAISON, Andreas KOPF, Edward YANG, Zachary DEVITO, Martin RAISON, Alykhan TEJANI, Sasank CHILAMKURTHY, Benoit STEINER, Lu FANG, Junjie BAI, and Soumith CHINTALA (2019), PyTorch: An imperative style, high-performance deep learning library, in Hanna M. WALLACH, Hugo LAROCHELLE, Alina BEYGELZIMER, Florence D'ALCHÉ-BUC, Edward A. FOX, and Roman GARNETT, editors, *Advances in Neural Information Processing Systems 32*, pp. 8024–8035, Curran Associates, Inc., doi:10.48550/arXiv.1912.01703.

Ida RAFFAELLI, Marko TADIĆ, Božo BEKAVAC, and Željko AGIĆ (2008), Building croatian wordnet, in Attila TÁNACS, Dóra CSENDES, Veronica VINCZE, Christiane FELLBAUM, and Piek VOSSEN, editors, *Proceedings of the 4th Global WordNet Conference (GWC 2008)*, pp. 349–359, Global WordNet Association, Szeged, Hungary, ISBN 978-963-482-854-9.

Josip SILIĆ and Ivo PRANJKOVIĆ (2005), *Gramatika hrvatskoga jezika*, Školska knjiga, Zagreb.

Octavia-Maria ŞULEA and Steve YOUNG (2019), Unsupervised inflection generation using neural language modelling, in Ignacio ROJAS, Gonzalo JOYA, and Andreu CATALA, editors, *Advances in Computational Intelligence*, pp. 668–678, Springer International Publishing, Cham, doi:10.48550/arXiv.1912.01156.

Krešimir ŠOJAT, Matea SREBAČIĆ, and Marko TADIĆ (2012), Derivational and semantic relations of Croatian verbs, *Journal of Language Modelling*, 0(1):111–142, ISSN 2299-8470, doi:10.15398/jlm.v0i1.34.

Richard Howard WICENTOWSKI (2002), *Modeling and learning multilingual inflectional morphology in a minimally supervised framework*, Ph.D. thesis, The Johns Hopkins University, `https://www.cs.swarthmore.edu/~richardw/pubs/thesis.pdf`.

Shijie WU, Ryan COTTERELL, and Mans HULDEN (2021), Applying the transformer to character-level transduction, in Paola MERLO, Jorg TIEDEMANN, and Reut TSARFATY, editors, *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1901–1907, Association for Computational Linguistics, Online, doi:10.18653/v1/2021.eacl-main.163.

Andrea ZIELINSKI, Christian SIMON, and Tilman WITTL (2009), Morphisto: Service-oriented open source morphology for German, in Cerstin MAHLOW and Michael PIOTROWSKI, editors, *State of the Art in Computational Morphology*, pp. 64–75, Springer Berlin Heidelberg, Berlin, Heidelberg, doi:10.1007/978-3-642-04131-0_5.

Domagoj Ševerdija
0000-0001-9501-1025
dseverdi@mathos.hr

Department of Mathematics,
University J. J. Strossmayer of Osijek
Trg Ljudevita Gaja 6
31 000 Osijek, Croatia

Rebeka Čorić
0000-0002-2388-385X
rcoric@mathos.hr

Department of Mathematics,
University J. J. Strossmayer of Osijek
Trg Ljudevita Gaja 6
31 000 Osijek, Croatia

Marko Orešković
0000-0002-3723-9256
moreskovic@nsk.hr

National and University Library
in Zagreb
Hrvatske bratske zajednice 4,
10000 Zagreb, Croatia

Lucian Šošić
0000-0002-1523-493X
luciansosic@gmail.com

Faculty of Humanities
and Social Sciences in Split
Poljička cesta 35,
21000 Split, Croatia

Domagoj Ševerdija, Rebeka Čorić, Marko Orešković, and Lucian Šošić (2024),
Detecting inflectional patterns for Croatian verb stems using class activation mapping, Journal of Language Modelling, 12(1):43–68
https://dx.doi.org/10.15398/jlm.v12i1.347