University of Passau
Department of Informatics and Mathematics

Project Report

A Corpus Reader and POS-Tagger for *MULTEXT-East* in *NLTK*

Alexander Böhm
Thomas Stieglmaier
Thomas Ziegler

October 15, 2015

Supervisor:
Behrang Qasemizadeh

Chair:
Digital Libraries and Web Information Systems
Prof. Dr. Siegfried Handschuh

This work presents a corpus reader for the Natural Language Processing Toolkit (***NLTK***) that is able to handle the ***MULTEXT-East*** corpus. The corpus features translations of the Book *1984* by Georg Orwell in many different languages supplemented with Part-of-Speech tags. This is quite unique in ***NLTK*** as most corpora are either available in only one single language while multilingual corpora do not containing Part-of-Speech tags. The native tagset of the corpus is the MSD tagset which allows very precise classification of words.
Additionally we evaluated multiple Part-of-Speech taggers based on ***NLTK*** and ***scikit-learn*** with a focus on how the performance of the taggers depend on the language and the tagset. In general more abstract tagsets can be handled better by the taggers we used, therefore the accuracy achieved with them is higher than with more complex tagsets.

Contents

1 Introduction **1**
- 1.1 Motivation . . . 1
- 1.2 Structure and Origin of This Work . . . 2

2 Background **3**
- 2.1 Natural Language Processing Toolkit . . . 3
- 2.2 Multext-East . . . 3
- 2.3 Part-of-Speech Tagging, Brill-Tagger . . . 4
- 2.4 Machine Learning, scikit-learn . . . 5
- 2.5 Tagsets . . . 6

3 Implementation **8**
- 3.1 MULTEXT-East Integration in NLTK . . . 8
  - 3.1.1 The MTECorpusReader . . . 9
  - 3.1.2 The MTETagConverter . . . 9
  - 3.1.3 The MTEDownloader . . . 10
  - 3.1.4 Sample Usage of our Corpus Reader Implementation . . . 10
- 3.2 Part-of-Speech Taggers for ***MULTEXT-East*** . . . 11
  - 3.2.1 The MTEBrillTagger . . . 11
  - 3.2.2 Part-of-Speech-Taggers with scikit-learn . . . 11
  - 3.2.3 Sample usage of our Part-of-Speech-Tagger Implementations . 13
- 3.3 Evaluation Framework . . . 14
  - 3.3.1 Sample usage of our Evaluation Framework . . . 14

4 Evaluation **17**
- 4.1 Evaluation Setup and Environment . . . 17
- 4.2 Metrics . . . 18
- 4.3 Results . . . 20
  - 4.3.1 Analysis per Tagger . . . 20
    - 4.3.1.1 ***NLTK***-Brill . . . 23
    - 4.3.1.2 ***scikit-learn***-MultinomialNB . . . 26
    - 4.3.1.3 ***scikit-learn***-Perceptron . . . 29
    - 4.3.1.4 ***scikit-learn***-LinearSVC . . . 32
    - 4.3.1.5 Conclusion for the Analysis per Tagger Evaluation . 34
  - 4.3.2 Analysis per Language . . . 35
    - 4.3.2.1 Czech . . . 36
    - 4.3.2.2 English . . . 37
    - 4.3.2.3 Estonian . . . 38
    - 4.3.2.4 Farsi . . . 39
    - 4.3.2.5 Hungarian . . . 40
    - 4.3.2.6 Polish . . . 41

4.3.2.7 Romanian 42
4.3.2.8 Serbian 43
4.3.2.9 Slovak 44
4.3.2.10 Slovenian 45
4.3.2.11 Conclusion for the Analysis per Language Evaluation 45
4.3.3 Raw Data 46

5 Restrictions and Challenges 48
5.1 Python 2, encoding issues with ***NLTK*** 48
5.2 Incomplete and incomparable tags in some ***MULTEXT-East*** resources 48
5.3 Conversion of the MSD Tagset to the Penn Treebank II Tagset 49

6 Conclusion 50

7 Appendix 51
7.1 Complete Tables of Evaluation 51

1 Introduction

1.1 Motivation

Up to now the *Natural Language Processing Toolkit* (***NLTK***) contains corpora in a wide variety of languages, but besides the corpus containing the *Universal Declaration of Human Rights* (UDHR Corpus) in over 300 languages most of them are monolingual. Additionally the UDHR corpus has no Part-of-Speech Tags[1] (POS-Tags) which makes it hard to use in deeper linguistic analyses.
In this work we present a corpus reader for the ***MULTEXT-East*** corpus in the ***NLTK*** project. It consists of POS-tagged versions of the Book *1984* by George Orwell in twelve different languages and is therefore an interesting addition to the currently available, mostly monolingual or untagged multilingual, corpora that can be used with ***NLTK***.
In addition to providing means to use the ***MULTEXT-East*** corpus in ***NLTK*** we evaluate different POS-taggers in ***NLTK*** and ***scikit-learn*** with a focus on how the taggers performance varies depending on the language and tagset.
Evaluating a POS-Tagger with a corpus containing the same text in several languages and comparing the results regarding accuracy, precision, recall and f1-score will lead to quite interesting results, such as the answer to the question *"Is the sentence structure important for the POS-Tagger?"*.

[1]For more information, see Section 2.3, or Chapter 1.7 of the Book *Natural Language Processing with Python* [BKL09].

1.2 Structure and Origin of This Work

The topic of this work was proposed by Behrang Qasemizadeh during the lecture *"Text-Mining Project"* at the University of Passau. In the proposal [ABZ15] we plan to focus on three points, the implementation of a ***NLTK*** corpus reader for ***MULTEXT-East***, a morphosyntactic analysis tool, and a POS-Tagger. The morphosyntactic analysis was left out, due to the huge amount of work done in the other two areas. This was also proposed by our supervisor.

In the following we give a short overview over the structure of this work:

- In the introduction, we give an overview over our topic
- and in the background, information about ***NLTK***, ***MULTEXT-East*** and POS-Tagging (related work, used algorithms) is provided.
- Then we show our implementation, consisting of an architectural overview about the implemented downloader, corpus reader, POS-Taggers and their usage.
- The implementation is followed by the evaluation, where we focus on the strengths and weaknesses of our POS-Taggers regarding the analysis of different languages.
- In the next section we focus on the restrictions and challenges during implementation and evaluation
- and in the end we conclude the work by summarizing our achievements.

The appendix consists of the references used in this thesis, the python documentation for the implemented corpus reader and POS-Taggers, and some additional tables, providing the full information of the POS-Tagger evaluation runs.

2 Background

2.1 Natural Language Processing Toolkit

NLTK [Bir06], is a flexible python framework for projects related to language processing and text mining. It contains various different corpora, such as the Brown Corpus[2] or WordNet [Mil95]. The UDHR corpus[3] is one of the few multilingual corpora in ***NLTK***, it comes without POS-Tags.
These corpora can be inspected and processed with pre-implemented methods and algorithms, that range from computing simple frequency distributions of words in a text to complete Part-of-Speech Taggers. For visualizing the findings ***NLTK*** allows to plot arbitrary graphs [4].
Penn Treebank II and *Universal Part-of-Speech Tagset* [PDM11] are the most commonly used tagsets inside ***NLTK***. While the Penn Treebank II tagset is quite precise and trimmed to English language only, the universal tagset is more abstract, applicable to many languages, and can therefore only provide coarse information about the type of a word in a sentence, in Section 2.5 more detailed information about the different tagsets can be found. A quite easy-to-implement POS-Tagger is the Brill-Tagger (c.f. Section 2.3) from ***NLTK***. It works with arbitrary tagsets, as well as different languages.

2.2 Multext-East

MULTEXT-East is a spin-off of the ***MULTEXT*** project which was made in the late 90's. ***MULTEXT-East*** is a corpus consisting of morphosyntactic information and POS-Tags of George Orwell's book *1984* for the languages Bulgarian, Czech, English, Estonian, Persian, Hungarian, Macedonian, Polish, Romanian, Slovak, Slovenian and Serbian [Erj]. The broad spectrum of available languages and the fact that there are POS-Tags and the lemma for each word in the book for each

[2] See `http://clu.uni.no/icame/manuals/BROWN/INDEX.HTM`.
[3] See `http://research.ics.aalto.fi/cog/data/udhr/`.
[4] Further use-cases and exemplary python code can be found in the first two chapters of the book *Natural Language Processing with Python* [BKL09]

included language makes the corpus quite interesting and unique.
The corpus is publicly available at the Slovenian homepage of the *Common Language Resources and Technology Infrastructure*[5]. The corpus contains a file for each language (`LL`) with annotations on word-level. They are named `oana-LL.xml`. Additionally there are files `oalg-LLMM.xml` which include the sentence alignments of the language `LL` to a language `MM`. Besides these files there are some xml header files and a folder containing the schema, used for encoding the xml files, it is compliant to the TEI P5 [Con] xml scheme. The basic structure of the `oana` files is as follows:

- At first there is a header, specific for each language, which contains an abstract, authors, copyright information and change notes,
- then, in the body of the xml file the annotated version of the book *1984* can be found. It is structured in paragraphs (`<p>`), sentences (`<s>`) and words / punctuation marks (`<w>`, `<c>`). Paragraphs as well as sentences and punctuation marks only have an id attribute, furthermore paragraphs contain one or more sentences and a sentence may contain words and punctuation marks. A word has a `lemma` and an `ana` attribute, where the lemma and the POS-Tag annotation of the word can be found. [6]

The tagset is feature-structure based and even more fine-grained than the Penn Treebank II tagset. It is represented by a string consisting of characters at certain positions, e.g. `#N` for a noun, unknown or inapplicable parts of tags can be symbolized by `-` or just left out if they only occur at the end, e.g. `#Npfs` (a `f`eminine `p`roper `N`oun in `s`ingular form (e.g. Julia)) is just more detailed than `#N`[7]. From the morphosyntactic descriptions in the tags also comes the name of the tagset: *MSD Tagset*. More about this tagset can be found in Section 2.5.

2.3 Part-of-Speech Tagging, Brill-Tagger

By determining the part of speech of a word in a given context, one gets useful information about the analyzed text. There are several approaches, supervised and unsupervised, to compute the POS-Tags automatically. The so gained information can for example be used for machine translation or information retrieval [Vou03]. One of the most commonly known supervised POS-Taggers is the Brill-Tagger [Bri92]. It is a rule-based part of speech tagger proposed by Eric Brill in 1992. During training,

[5] It is available under *CC BY-NC-SA 4.0* license [EBD+10].
[6] There are two languages, Bulgarian and Macedonian, whose xml scheme differs from the presented structure, more on that can be found in Section 5.
[7] All possible tags are listed language-wise on `http://nl.ijs.si/ME/V4/msd/html/index.html`.

the tagger first uses an initial tagger to generate tags. In our implementation we use an unigram-tagger for this purpose. Afterwards the tagger tries to find rules that transform the tag from the initial tagger into the correct tags. These rules are created based on a set of templates. After training, the tagger saves the rules to use them for tagging a corpus.
When the tagger tags an untagged corpus, it first uses the initial tagger and then transforms the tags according to the previously learned rules.

2.4 Machine Learning, scikit-learn

Machine learning describes the process of algorithmic knowledge generation by analyzing a given set of training data. The deducted rules have to be abstract enough to cover new data, otherwise the algorithm is over-approximating.

***scikit-learn* [PVG+11]** is a machine learning framework for Python and built on NumPy, SciPy and matplotlib. It is open source and BSD-Licensed. It offers a wide range of mechanisms and helpers for classification, regression, clustering and dimensionality reduction. Part-of-Speech tagging is a multi class classification task. This has to be supported by the classifiers.

Naive Bayes classifiers are based on a naive implementation of the Baysian-Theorem. Such a classifier assumes that all attributes are independent of each other. A multinomial naive Bayes is useful where the frequencies have been generated by a multinomial, such as in natural language processing. It's functionality for an implementation in ***scikit-learn*** is described in related literature [MRS08]. McCallum and Nigam analyse the performance of a multi-variant Bernoulli model and a multinomial one [MN98]. Their result is that a multinomial one, compared to the Bernoulli model, provides an average reduction of 27,% in error. In training and classification a baysian classifier has a complexity of $O(n_{features} \cdot n_{samples})$ [ZW05].

Support Vector Machines based classifiers represent the data as vectors within a n-dimensional vector room. The task is to find a hyperplane which separates the vectors in such a way that the margin between the hyperplane and the vectors is maximal [Ber]. If it is not possible to find a hyperplane which separates the data in the expected way, a so called kernel-trick is used: The vector room will be extended to a higher dimension until a sufficient hyperplane is found [Lin]. For a high amount of features this task is quite complex (a problem of quadratic programming) and thus the complexity is between $O(n_{features} \cdot n_{samples}^2)$ and $O(n_{features} \cdot n_{samples}^3)$ [CL11].

Perceptrons are simplified artificial neuronal networks which can be used as linear classifier [Roj96]. Within this network, each neuron is assigned a specific weight which will be honored when calculating the output. Training of a perceptron is mostly done by using backpropagation: At first the input data is propagated trough the neuronal network, then the difference of the expected to the current output is compared. This difference is propagated trough the network in the reversed order to adjust the weight of the neurons. This will result in a reduction of their influence on this specific input [Kri07]. This process can be repeated to adjust the performance of the classifier. Those repetitions make the training of the classifier more complex (in terms of computational complexity), but the approach is faster for testing. Neuronal networks can be used as alternatives to conditional random fields or maximum-entropy models without performance loss [Col02].

2.5 Tagsets

There are different tagsets that can be used for part of a speech tagging. In this chapter we will describe the two different tagsets we used.

The **MSD Tagset** is the tagset originally used in the ***MULTEXT*** corpus. It is a very fine-grained tagset that allows one to tag words very precisely.
A tag in this tagset usually starts with an '#', is followed by one upper case character that indicates the category of the word and a number of lower case characters or digits that indicate attributes. The number of attributes can depend on the category of the word and the language.
For example a noun is tagged with an 'N' which indicates the category `noun`. In English it is followed by three attributes, the type ('c' for a common noun, 'p' for a proper noun), the gender and the number. In Romanian on the contrary it has six attributes. The first three are the same as in English while the following attributes indicate the case, definiteness and if it is a clitic.
To keep the tags as uniform as possible while still being able to take subtleties of different languages into account it is possible to omit attributes by using a dash in its place. The dash indicates that the property of the word is undecidable in the current language. After an arbitrary number of dashes the tag can continue assigning values to attributes. Dashes in the end of a tag can be omitted.
Lets consider 'winner', an English noun that has no gender. It would be tagged `#Nc-s` as it is a common noun in singular. The word 'German' (meaning the language) would be tagged `#Nc` because the attributes gender and number are undecidable.
Additionally to the MSD tags, ***MULTEXT-East*** provides less specific POS-Tags

and their mapping to the MSD tags in the file `msd2ctag.tbl`. The development of this mapping as been stopped, so the mapping is incomplete for almost all languages.

The **Universal Tagset** is a much simpler tagset than the MSD tagset. It was introduced by Slav Petrov et al. [PDM11]. It features only basic tags that indicate the category as indicated by the first character in the MSD tagset. To be able to use this tagset a method mapping MSD tags to Universal tags was implemented by a mapping from MSD category to Universal tag.

The **Penn Treebank II Tagset** is more sophisticated than the Universal tagset but lacks the flexibility of the MSD tagset. It doesn't have a modular system like the MSD tagset but fixed tags for different kinds words. So one can distinguish between proper and common nouns and the number is taken into account, but not the gender. The tagset may be handy to tag corpora in English but it obviously can't model all subtleties of all the other languages in the ***MULTEXT*** corpus like Romanian or Farsi.

3 Implementation

In this section we describe our integration of ***MULTEXT-East*** in ***NLTK*** and the on top implemented POS-Taggers. Due to problems with Unicode in Python 2 we were forced to support Python 3 but as both versions are in use and Python 2 is not really deprecated we decided to write the code in a way that it can be used with both versions of Python. As ***NLTK*** version 2 does not support Python 3 we were forced to use ***NLTK*** 3. The older version is not supported due to incompatible interfaces and implementations, for example those of the Brill Tagger.
The additional POS-Tagger implemented in ***scikit-learn*** will also be discussed in this section.

Sourcecodes can be obtained from our GitHub-Repository at `https://github.com/jwacalex/MULTEX-EAST-PoS-Tagger`. For more information about the sourecodes and tools we provide there, please refer to this report, the python documentation and the `README.md`-files within the repository.

3.1 MULTEXT-East Integration in NLTK

In order to be able to use the ***MULTEXT-East*** corpus in ***NLTK*** like every other corpus, we implemented a custom `CorpusReader` and some other tools which provide convenient features like the conversion of the MSD tagset to the Universal tagset, and a downloader which handles the installation of the ***MULTEXT-East*** sources on the users' computer. While the corpus reader and the tagset conversion class reside in the file `mte.py`, the downloader is separate in a file called `MTEDownloader.py`. Since commit `c6e7a6d5be33308c3c92e531124e93cd7f1908c4` in the ***NLTK*** repository at github[8] The corpus reader is shipped with ***NLTK*** and does not have to be downloaded separately. In the time we write this report, it is also worked on integrating the ***MULTEXT-East*** sources into ***NLTK***s own download tool.

[8] See `https://github.com/nltk/nltk`.

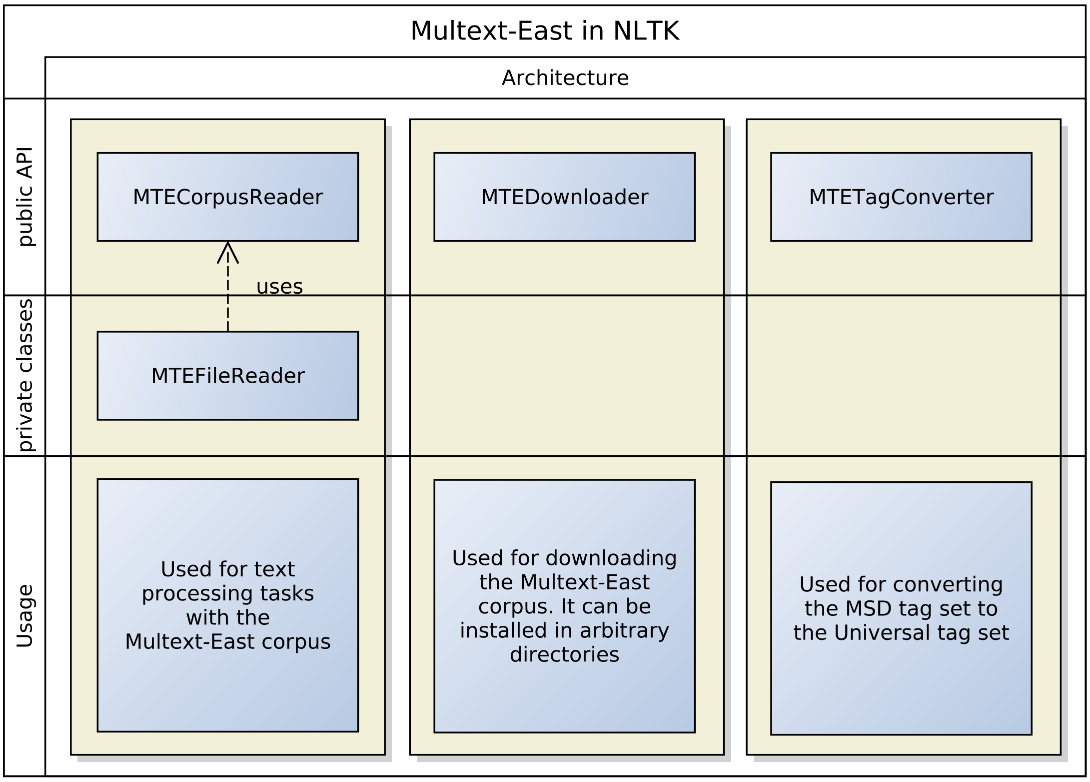

Figure 3.1: Overview over the architecture of the corpus reader for ***MULTEXT-East*** inside ***NLTK***

3.1.1 The MTECorpusReader

The `MTECorpusReader` extends the `TaggedCorpusReader` class of ***NLTK***. As constructor argument one can choose which languages of the ***MULTEXT-East*** corpus should be loaded. As all other corpus readers, the words, sentences, paragraphs, tagged words, and tagged sentences can be retrieved. Additionally we have methods that return the lemmatized words and sentences.
For all methods that return something with tags a filter can be given as parameter such that e.g. only words with `#N-s` are in the returned list. The filter works in the way, that all `-` are seen as unspecified and can therefore have an arbitrary value, also if the given filter is too short it is filled up with `-` to the needed length internally. This means that with the example filter, for the English file, we would retrieve all singular nouns not regarding differences in gender or if the noun is proper or not.

3.1.2 The MTETagConverter

The `MTETagConverter` class contains the method for converting MSD tags into the matching Universal tags[9]. If any other tagset is required a mapping file that contains

[9] A mapping for MSD to Universal tags is predefined by us and can be used out of the box

a mapping from the MSD tagset to the target tagset can be added.
In this case it is important that one MSD tag is mapped to no more than one target tag. But multiple MSD tags can still map to the same target tag.

3.1.3 The MTEDownloader

The `MTEDownloader` is a standalone download manager to obtain the files if it is not possible to use the ***NLTK*** functionality. It can be either started via executing the script as a python program (it has a main method) or by directly calling `MTEDownloader.download()`. At first one has to choose the installation directory, then the corpus is downloaded from `clarin.si` and extracted.

3.1.4 Sample Usage of our Corpus Reader Implementation

The following code shows some basic examples how the corpus reader, the Downloader and the provided utility methods could be used:

```
> # at first import all necessary files
> import MTEDownloader, mte
>
> # then (if not yet done) download the Multext-East corpus
> # this could also be done via the nltk downloader nltk.download()
> MTEDownloader.download()
Where should the corpus be saved?[(0, '/home/stieglma/nltk_data'), (1, '/usr/share/nltk_data'),
     (2, '/usr/local/share/nltk_data'), (3, '/usr/lib/nltk_data'), (4, '/usr/local/lib/nltk_data'
     ), (5, 'custom')] [0]: 0
Downloaded 14800805 of 14800805 bytes (100.00%)
Download finished
Extracting files...
Done
>
> # if you do not have an nltk version where our corpus reader is already
> # integrated, you have to manually create it
> # now we open the English version of the book 1984 with our reader
> reader = mte.MTECorpusReader(root="/path/to/multext/corpus/", fileids=['oana-en.xml'])
>
> # otherwise you can just do the following
> from nltk.corpus import multext_east as reader
>
> # and then we retrieve the first word in the first word/tag tuple of this file
> reader.tagged_sents(fileids="oana-en.xml")[0][0]
('It', '#Pp3ns')
>
> # the tag is now in the Multext-East (MSD) format, we want it to be
> # the more well known corresponding universal tag:
> reader.tagged_sents(fileids="oana-en.xml", tagset="universal")[0][0][1]
'PRON'
>
> # now we want to see something in the concordance view:
> from nltk import Text
> Text(reader.words(fileids="oana-en.xml")).concordance("brother")
Displaying 2 of 80 matches:
ollow you about when you move . Big Brother is watching you , the caption benea
se-front immediately opposite . Big Brother is watching you , the caption said
```

3.2 Part-of-Speech Taggers for *MULTEXT-East*

For the evaluation of our work we implemented Part-of-Speech taggers based on different algorithms, on the one hand we have Brill's algorithm, which is also shipped with ***NLTK*** and on the other hand we have three machine learning algorithms that are shipped with ***scikit-learn***. The following sections will provide an overview over the implementations and show the way how they are used.

3.2.1 The MTEBrillTagger

The implementation contains a wrapper around ***NLTK***s implementation of the Brill Tagger. It builds a Brill Tagger based on a default tagger which can be specified. By default a unigramm tagger is used. Additionally it contains a method to evaluate the tagger.
The Brill Tagger for the ***MULTEXT-East*** corpus can be configured like the standard ***NLTK*** Brill Tagger. Additionally the set of templates can be specified. Either a function from `nltk.tag.brill` returning a list of tags (e.g. `fntbl37`) can be specified as a string or a list of templates can be passed in the `template` parameter.

The **MTEBrillTagger** needs a set of tagged sentences to train the tagger. It is possible to tune the behavior of the tagger by giving additional parameters like the maximum number of rules or a different initial tagger.

The **evaluate** method takes a set of test sentences to evaluates the already trained tagger. It prints the accuracy.

The **metrics** method takes a set of sentences for testing and gives deeper information about the evaluation. There are accuracy, precision, recall, f-score and out of vocabulary words. Additionally a confusion matrix can be generated.

To evaluate the Brill Tagger there is the class **BrillTaggerEval**. It takes a whole corpus, tagged with MSD tags and will do the ten-crossfold-validation as described in section 4. It also takes a string specifying the tagset as well as a few parameters to tune the Brill Tagger. The **evaluate** method can do an n-crossfold-validation while the default for `n` is ten. The output will be the averages of the values specified in the metrics method of the MTEBrillTagger. Confusion matrices are not supported by this method at the moment.

3.2.2 Part-of-Speech-Taggers with scikit-learn

As a conclusion of the performance findings described in Section 2.4 we chose a multinomial naive Bayes, a support vector machine with a linear kernel and a perceptron

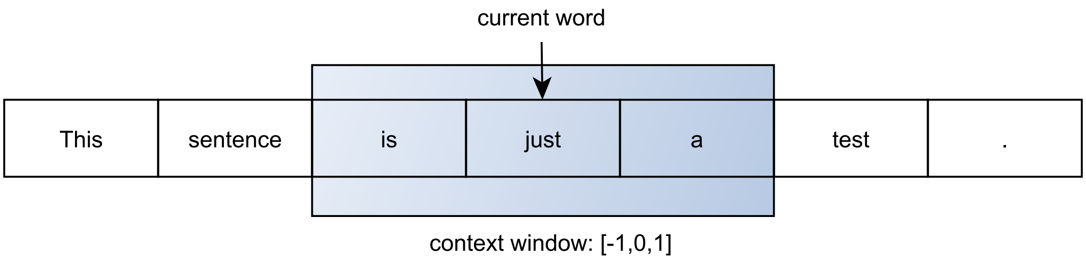

Figure 3.2: Application of the Context Window to a Sentence

as classifiers for our implementation. For the implementation itself we have chosen `MultinomialNB`[10], `LinearSVC` [11] and `Perceptron`[12]. All classifiers are configured with the default configuration.

Loading the Corpus is done by using `MTECorpusReader` for a given language. Our tagger uses the default ***NLTK***-format or tagged sentences, so it can be easily changed to any other corpus.

Conversion of the Tagset is optional but for comparison we decided to use the *Universal* tagset and the default MSD tagset. The conversion itself is performed by a method within `MTETagConverter`.

Splitting the Data is performed by our own implementation for a 10-fold-cross validation. For details, please refer to section 4.

Applying the Context Window to the tagged sentences will result in the training vectors. The context window is specified as a list of positions, relative to the currently processed word within a sentence. This results in a dataset which maps the words, their position and tags within the context window to the tag of the currently processed word. An example can be found in Figure 3.2.

[10] `http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html#sklearn.naive_bayes.MultinomialNB`

[11] `http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html#sklearn.svm.LinearSVC`

[12] `http://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Perceptron.html`

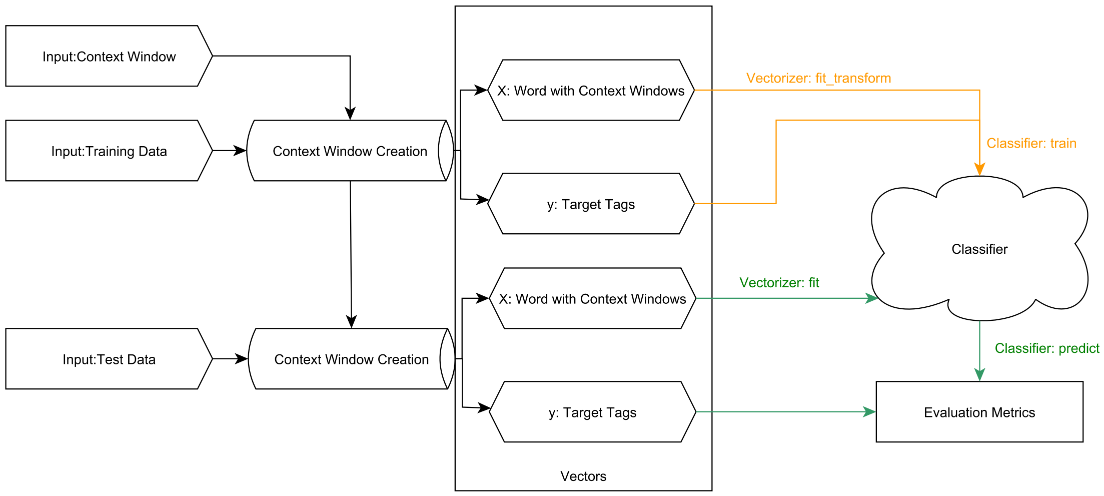

Figure 3.3: ***scikit-learn*** Part-of-Speech Tagger Overview

Vectorization is handled by `DictVectorizer` [13], which vectorizes the data by a binary one-hot coding. It performs the different transformations for training- and test-data as described in the documentation.

Training the Classifier and Predicting the Results is performed by calling the `fit` and `predict` method of the classifier. The first functions generates a model corresponding to the abstraction of the provided data where the second performs the classification of new, unknown data.

3.2.3 Sample usage of our Part-of-Speech-Tagger Implementations

```
> from MTEPosTaggers.MTEBrillTagger import MTEBrillTagger
> from MTEPosTaggers.MTESKTagger import MTESKTagger
> from MTEPosTaggers.MTESKTagger import *
> from mte import MTECorpusReader


> loaded_corpus = MTECorpusReader(root="/path/to/multext/files/", fileids="oana-en.xml") #Loading
    English Language
> tagged_sents = loaded_corpus.tagged_sents() #get the tagged sents


> brill = MTEBrillTagger(tagged_sents) #instanciate the MTEBrillTagger with default settings

#omitted, training output brill tagger

> sktagger = MTESKTagger(tagged_sents,context_window_generator=around1()) #instance the
    MTESKLeanTagger with default settings (MultinomialBayes, context window is one word left and
    right from the current word and itself)

> brill.evaluate(tagged_sents) #evaluate the model
0.9660261204763823

> sktagger.evaluate(tagged_sents) #same for the scikit tagger
0.8907907101624379
```

[13] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.DictVectorizer.html#sklearn.feature_extraction.DictVectorizer

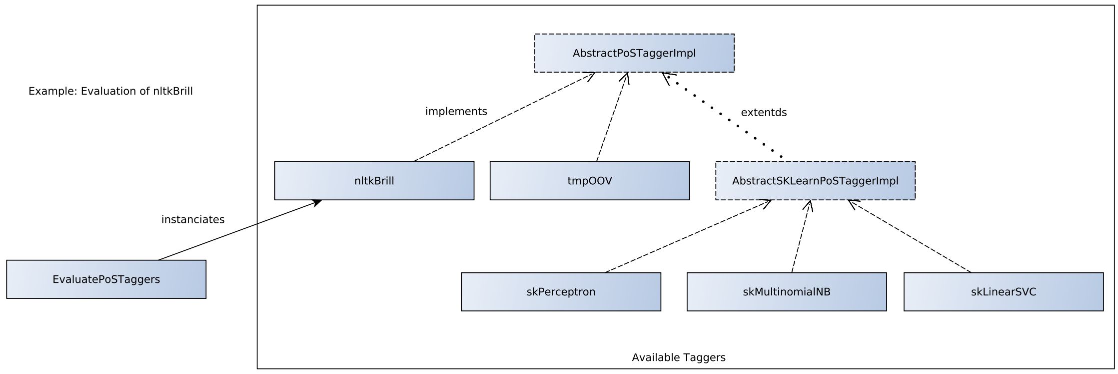

Figure 3.4: Evaluation Framework Architecture

3.3 Evaluation Framework

We decided to write a helper to perform the evaluation because there is a high amount of possible combinations: languages × tagsets × taggers × configuration options. For each of this combination it is necessary to perform the n-folds.

Each tagger is represented by a class which implements `AbstractPoSTaggerImpl`. The abstract class contains an **`evaluate`** method and a public list named **`config options`**. This list contains the different configurations for the tagger. During the evaluation run, all options will be benchmarked.

The evaluate method will be called for each combination of language, tagset and configuration option. As parameter it will get the tagged sentences for training and testing and the metric which should be returned as result.

Each implementation of a tagger ensures itself that the returned results are comparable, because the framework does not calculate the metrics itself. For our implementation of taggers we convert the results to the ***NLTK***-format and return the calculated metrics by the `nltk.metrics.scores`[14] module.

For the evaluation we use a 10-Fold cross validation where the minimum, maximum and averaged value (with standard deviation) is stored in the results file.

3.3.1 Sample usage of our Evaluation Framework

To use the framework, each tagger has to meet the requirements stated above and be instanciatable from the `EvaluatePoSTaggers` script.
Prior to execution the following variables must be checked and if necessary, changed.

- `result_file`: Location of the results file.

[14]`http://www.nltk.org/_modules/nltk/metrics/scores.html`

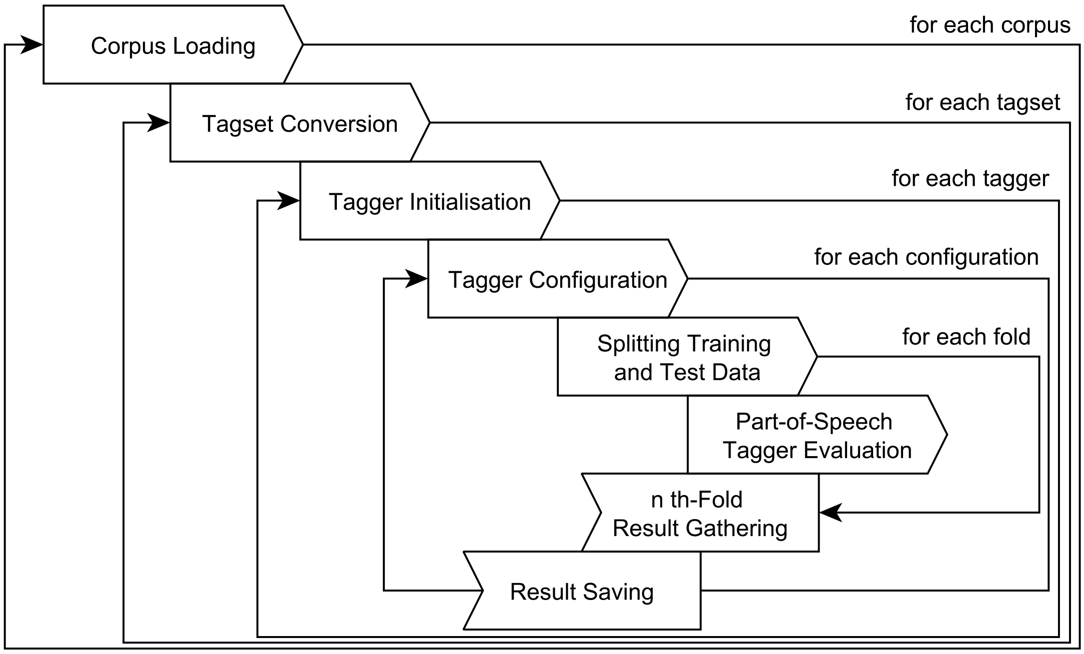

Figure 3.5: Evaluation Pipeline Overview

- `languages`: List of languages to evaluate.
- `corpus_root`: Root directory of the ***MULTEXT-East***-Corpus.
- `taggers`: List of taggers to evaluate.
- `metrics`: List of metrics to evaluate. Must be implemented by the `get_result` function of each tagger.
- `n_folds`: Number of folds for cross validation (min: 2)
- `n_workers`: Number of workers for parallel execution (min: 1, max `n_folds`)

The following commented snipped shows how to use our brill-tagger within the framework, it can be adapted for any other tagger:

```
class nltkBrill(AbstractPoSTaggerImpl):
    config_options = ['fntbl37'] #define the configurations to evaluate
    max_rules = 250
    min_score = 2
    min_acc = None
    results = None

    def evaluate(self, sents_train, sents_test, config_option):
        # start benchmarking training time
        t = time()

        # instanciation of the tagger + training
        brill = MTEBrillTagger(sents_train, max_rules=self.max_rules, min_score=self.min_score,
            min_acc=self.min_acc, template=config_option)

        # stop timekeeping for training
        self.training_time = time() - t
        t = time()

        # run tagging and store the results within instance
        self.results = brill.metrics(sents_test, printout=False)
        self.prediction_time = time() - t
```

```
        return self

    def get_result(self, metric):
        # return the corresponding results by
        # getting the right values out of the self.results
        # structure which is defined by MTEBrillTagger

        if metric == 'accuracy':
            return self.results[0]
        elif metric == 'precision':
            return self.results[1]
        elif metric == 'recall':
            return self.results[2]
        elif metric == 'f1':
            return self.results[3]
        elif metric == 'training_time':
            return self.training_time
        elif metric == 'prediction_time':
            return self.prediction_time
        elif metric == 'oov':
            return -1
```

4 Evaluation

For evaluating the performance of the taggers we use a 10-fold-cross validation approach. The data is split into ten, approximately equal, parts. In each run another part is used for testing the tagger while the rest is used for training. The results are averages of the ten individual runs.

4.1 Evaluation Setup and Environment

In the evaluation we used the corpus reader and the taggers implemented by us that were described in former sections. The complete evaluation was run with Python 3.4 due to encoding issues we encountered with earlier versions. ***NLTK*** was used with version 3.0.4 and ***scikit-learn*** in version 0.16.1.

For our Brill-Tagger implementation we used four template sets which are pre-defined by ***NLTK***:

- nltkdemo18
- nltkdemo18plus
- fntbl37[15]
- brill24[16]

The context windows used for the ***scikit-learn***-based taggers are not pre-defined but specified by us, there may be others or even combinations of context windows such that the taggers would perform better, but testing more context windows or their combinations was not possible within the scope of this task. Therefore we focused on the ten most common windows, from one to three words left, right or around from the current word (plus the current word) and only the current word. An overview over all configurations and their abbreviations used later on can be seen in Table 4.1.

The evaluation was run on a computer with 32 GB of RAM and an Intel(R) Core(TM) i7-4771 CPU @ 3.50GHz with our evaluation framework described earlier.

[15] based on the fntbl-distribution: `http://www.cs.jhu.edu/~rflorian/fntbl/index.html`
[16] based on the work of Eric Brill [Bri95]

4.2 Metrics

Accuracy, precision, recall and f1-score are measured by the appropriate methods within ***NLTK***. They calculate the micro averaged values: The global results of the false positive (f_p), false negative (f_n) and true positive (t_p), true negative (t_n) are calculated by summing up those values for each class [Seb02].
When we talk about the *best* performance of a tagger we always think of the f1-score.

Accuracy is the number of true results divided by the number of all data:

$$\frac{t_p + t_n}{t_p + t_n + f_p + f_n}$$

If accuracy is 1.0 all predicted values are the same as the gold standard.

Precision is the number of true positives (items correctly labeled as belonging to the positive class) divided by the total number of elements belonging to the positive class:

$$\frac{t_p}{t_p + f_p}$$

If precision is 1.0 the classifier has labeled all data (for this class) correctly.

Recall is the number of true positives divided by the total number of elements that belong to the positive class:

$$\frac{t_p}{t_p + f_n}$$

If recall is 1.0 the classifier as labeled all data as belonging to the right class. But recall does not contain any information about the number of data that is labeled as the wrong class (false positives)

F1-Score describes the harmonic mean of precision and recall and can be seen as weighted averaged between those two metrics:

$$F_1 = 2 \cdot \frac{\text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}$$

Out-of-Vocabulary words is the number of words, that appear in the test corpus but not in the training corpus and thus are *new* to the tagger. This number does not change while evaluating with different tagsets as the partitions of training- and test-sets in the cross-fold validation does not depend on the tagset but only on the outline of the corpus.

Spearman's rank correlation coefficient [17] is a non-parametric measure of the statistical (in)depence of two variables. It answers to the question "how well can a relationship between those to variables described by a monotonic function". If the data fits perfectly to a monotonic function, the result itself will be -1 or 1.

The coefficient itself is defined as the Pearson correlation coefficient between the ranked data points. It is calculated for each data point x_1 and y_1, with $d_i = x_i - y_i$ the difference between the ranks and n the number of data point tuples.

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

The correlation can be interpreted as the association between the independent variable X and the depended one Y. If both variables increase the ρ, the sign of the Spearman correlation coefficient is positive, but if Y decreases while X increases ρ is negative. If there is no tendency that Y will increase or decrease when X is increasing, the coefficient ρ is zero. This is interpreted as a very low correlation between the data points.

[17] Further reading [Spe04]

Configuration	Abbreviation	Description
baseline	base	Baseline (NLTK Unigram Tagger)
[0]	base	Baseline (Current word)
brill24	br24	Brill 24
fntbl37	ftbl37	Fnt Tables 37
nltkdemo18	nl18	NLTK Demo Ruleset
nltkdemo18plus	nl18+	NLTK Demoe Rulset
[-3, -2, -1, 0, 1, 2, 3]	arnd3	Three words
[-2, -1, 0, 1, 2]	arnd2	Two words around
[-1, 0, 1]	arnd1	One word around
[-1, 0]	left1	One word left from
[-2, -1, 0]	left2	Two words left
[-3, -2, -1, 0]	left3	Three words left
[0, 1]	right1	One word right
[0, 1, 2]	right2	Two words right
[0, 1, 2, 3]	right3	Three words

Table 4.1: Mapping of the Name of Configuration Options to Abbreviations

Language	Minimum	Maximum	Average	Standard Deviation
Czech	15.3895	16.9205	15.8649	0.4397
English	4.3627	5.1983	4.7758	0.2470
Estonian	15.3014	17.2707	16.2746	0.6795
Farsi	5.3365	6.9215	6.1749	0.4515
Hungarian	16.3756	18.5125	17.5267	0.5290
Polish	16.9517	18.2349	17.7253	0.4454
Romanian	8.2533	8.9866	8.6080	0.2502
Slovak	15.3369	17.0994	16.4197	0.5002
Slovenian	12.2008	13.3773	12.8194	0.4113
Serbian	12.0504	13.7943	13.0699	0.5891

Table 4.2: Out of Vocabulary Words per Language

4.3 Results

In this section we will have a short look at each implemented tagger and each ***MULTEXT-East*** language. We will analyze the most significant parts of the results in this section. Information about the raw data can be found in Section 4.3.3. For better readability of tables and plots we decided to create abbreviations for the tested configurations. The mapping of configuration to abbreviation and a description of the configuration can be seen in Table 4.1.

When it comes to out-of-vocabulary statistics – which are independent of configuration and used tagger – the differences are huge. Where in the Czech version approximately 15.8 % words are not known from the training set, in English only 4.7 % are out-of-vocabulary. This has to be taken into consideration when drawing conclusions about the performance of the taggers on different languages. In the following sections we will at first compare all languages per tagger and afterwards compare all taggers per language.

4.3.1 Analysis per Tagger

In this section we will analyze and discuss our findings per Tagger. Thus we look at each tagger in detail without comparing them, but instead comparing the taggers performance on different languages and also different configuration options. For a better overview, the tables and plots in Figures 4.3 4.1 4.4 and 4.2 can be used.

Universal Tagset While tagging the parts-of-speech with the *universal* tags, the `skLinearSVC` is the best tagger overall. However, the context windows which is used while tagging differs.
While Farsi can be tagged best with the context window that only looks at the

Tagger	Configuration	Language	Accuracy (second line: standard deviation)	F1-Score (second line: standard deviation)	Out of Vocabulary
skLinearSVC	[0]	Farsi	0.9376 0.0015	0.8874 0.0039	0.0617
skLinearSVC	[-1, 0]	English	0.9316 0.0018	0.8590 0.0067	0.0478
skLinearSVC	[0, 1, 2]	Romanian	0.9179 0.0029	0.8452 0.0052	0.0861
skLinearSVC	[-1, 0]	Estonian	0.9067 0.0032	0.8271 0.0050	0.1627
skLinearSVC	[-1, 0, 1]	Polish	0.9042 0.0029	0.8252 0.0047	0.1773
skLinearSVC	[-1, 0, 1]	Slovenian	0.9170 0.0041	0.8215 0.0087	0.1282
skLinearSVC	[-1, 0]	Hungarian	0.8985 0.0040	0.8188 0.0093	0.1753
skLinearSVC	[-1, 0, 1]	Serbian	0.8817 0.0034	0.8167 0.0058	0.1307
skLinearSVC	[0, 1]	Czech	0.8908 0.0031	0.7981 0.0055	0.1586
skLinearSVC	[0, 1]	Slovak	0.8722 0.0039	0.7804 0.0063	0.1642

Table 4.3: Best Results over all Taggers with the Universal Tagset

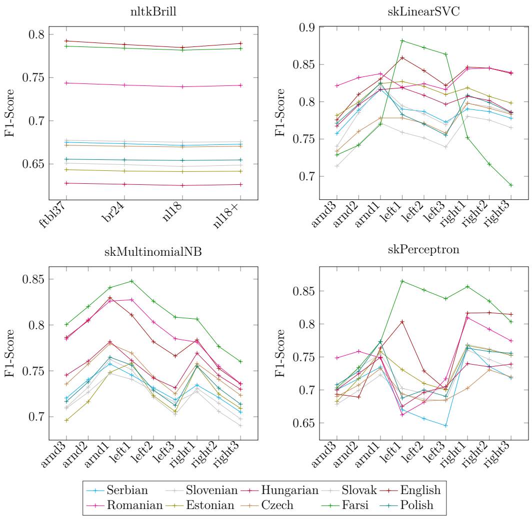

Figure 4.1: Chart of Results for the Universal Tagset over all Taggers and Configurations

current word (also called baseline in our evaluation) all other languages (besides Romanian) can be tagged best with either looking at the word before (`left1`), after (`right1`) or both (`arnd1`) together with the current word.
`arnd1` and `left1` are the configurations that can tag the most languages best, after that, `right1` is following, and the last one is `right2`.

The performance differences when using different configurations only effects the ***scikit-learn***-based taggers. For the `Brill`-Tagger almost no difference can be measured. The performance differences with different configuration while using ***scikit-learn***-based taggers can be seen for each of the taggers. However there are some remarkable differences: While `skLinearSVC` tags different languages best (regarding the f1-score) with different configurations, for `skMultinomialNB` always Farsi is the language with the highest f1-score and according to that the most other languages always have the same order when looking at the f1-score for the different configurations.

While `nltkBrill` performs best on English (with a f1-score, little less than 0.8) and then Farsi, for `skLinearSVC` the language that can be handled best is Farsi (with a f1-score of approximately 0.89) end then English. So the taggers are not only different from the f1-score but also from the order of which language could be tagged best. Deeper insights will be provided by spearman-correlation matrices that come in later sections.

Tagger	Configuration	Language	Accuracy (second line: standard deviation)	F1-Score	Out of Vocabulary
skLinearSVC	[-1, 0]	English	0.8914 0.0038	0.8044 0.0080	0.0478
skLinearSVC	[-1, 0]	Farsi	0.8483 0.0054	0.7709 0.0117	0.0617
skLinearSVC	[-1, 0, 1]	Romanian	0.8685 0.0037	0.7666 0.0079	0.0861
skLinearSVC	[0, 1]	Hungarian	0.8269 0.0045	0.6652 0.0103	0.1753
skLinearSVC	[0, 1]	Estonian	0.7714 0.0063	0.6439 0.0087	0.1627
skLinearSVC	[-1, 0]	Slovenian	0.7801 0.0049	0.6027 0.0074	0.1282
skLinearSVC	[-1, 0]	Serbian	0.7343 0.0065	0.5881 0.0122	0.1307
skLinearSVC	[0, 1]	Slovak	0.7191 0.0073	0.5767 0.0094	0.1642
nltkBrill	brill24	Czech	0.7324 0.0057	0.5737 0.0062	0.1586
skLinearSVC	[-1, 0]	Polish	0.6991 0.0050	0.5668 0.0085	0.1773

Table 4.4: Best Results over all Taggers with the MSD Tagset

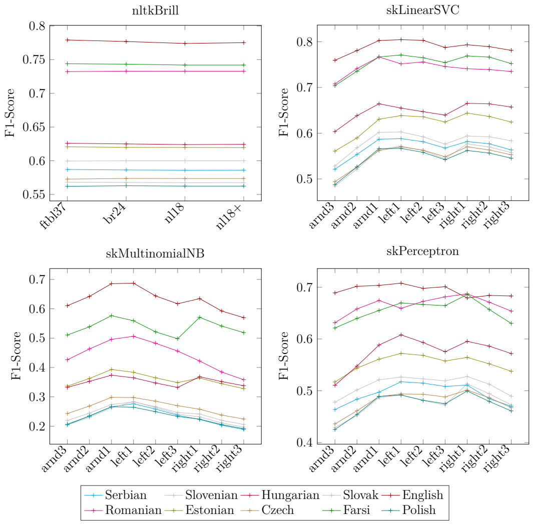

Figure 4.2: Chart of Results for the MSD Tagset over all Taggers and Configurations

MSD Tagset While tagging the parts-of-speech with the *MSD* tags, the `skLinearSVC` is the best tagger overall, only for Czech, the `nltkBrill` tagger with the `brill24` template set performs better. Equal to the *universal* tagset outcomes, the context windows which performed best with `skLinearSVC` were `left1` with 5 occurrences and `right1` with 3 occurrences followed by `arnd1` which occurs once. When looking at the Tables 4.3 and 4.4 at once it can be seen that the choice of the context windows should not only be based on the grammar of the tagged language, but also on the used tagset.

What also can be seen is, that as well as for the *universal* tags, the performance of `nltkBrill` is rather constant when comparing the different configurations. A thing that changes from the *universal* to the *MSD* tagset, is the order of the languages regarding f1-score. So not only the grammar of the language is playing a role in the difficulty of tagging a language, but also the used tagset.

In contrast to the evaluation results for the *universal* tagset for the ***scikit-learn***-based taggers, the order of the languages regarding f1-score is with the *MSD* tagset not that non-deterministic. Instead all configurations perform almost at the same level for a certain language. But still, `skLinearSVC` is the best tagger out of the ***scikit-learn***-based taggers. The other two, `skMultinomialNB` and `skPerceptron` have a maximal f1-score of approximately 0.7 for English, the f1-score of `skLinearSVC` is about 0.1 higher.

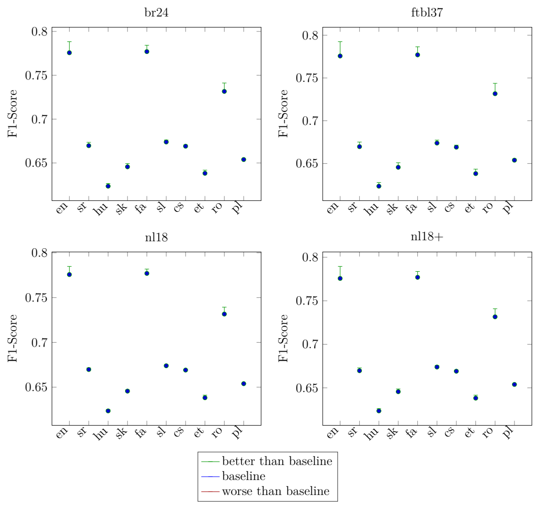

Figure 4.3: Comparison over all Languages and Configuration Options for the nltkBrill Tagger within the Universal Tagset

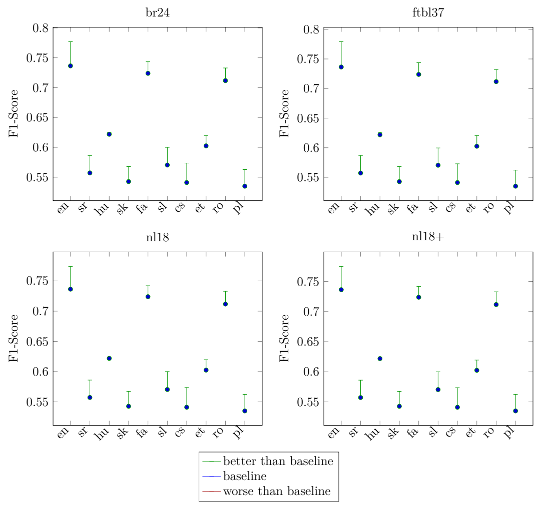

Figure 4.4: Comparison over all Languages and Configuration Options for the nltkBrill Tagger within the MSD Tagset

4.3.1.1 *NLTK*-Brill

In this section we present the results, achieved with the Brill Tagger. Figures 4.3 and 4.4 show the relative improvements of using the Brill Tagger compared to a Unigram Tagger which is used as baseline. It can be seen that with all languages and template sets, the Brill Tagger strictly performs better than the baseline. When looking at the two figures separately, almost no difference between the four plots ca be found for both tagsets. It can also be seen that the f1-score improvement from using the Brill Tagger instead of the Unigram Tagger is much higher for the *MSD* Tagset. Therefore we can say, that for more complicated tagsets the use of more sophisticated taggers has more impact than for easier, more abstract, tagsets. But also the language and the percentage of out-of-vocabulary words influence the performance of the tagger.

In Table 4.5 an overview over the performance of the Brill Tagger is provided. For every language the best result according to the f1-score for the MSD and Universal tagset is shown as well as the number of out-of-vocabulary words. The name of the template set which was used to gain the results is also provided.
For every tagset the highest values are marked in green, the lowest in red.

Best results for a corpus tagged with the MSD tagset were achieved for the English version with an f1-score of 0.7792. The English corpus also has the lowest amount of out-of-vocabulary words with 4.8 %. An f1-score of 0.5629 was the worst result. It was achieved with the Polish corpus which has 17.7 % of out-of-vocabulary words. Throughout the results corpora with a low number of out-of-vocabulary words are

Language	Configuration	Tagset	Time (seconds) Training	Tagging	Accuracy (second line: standard deviation)	F1-Score	Out of Vocabulary
Czech	fntbl37	Universal	26.2	7.82	0.8296 0.0047	0.6716 0.0075	0.1586
	brill24	MSD	79.5	10.53	0.7324 0.0057	0.5737 0.0062	
English	fntbl37	Universal	105.7	15.35	0.9263 0.0023	0.7924 0.0080	0.0478
	fntbl37	MSD	124.4	14.16	0.9087 0.0024	0.7792 0.0065	
Estonian	fntbl37	Universal	32.4	7.57	0.8209 0.0068	0.6434 0.0109	0.1627
	fntbl37	MSD	69.9	8.74	0.7793 0.0066	0.6207 0.0104	
Farsi	fntbl37	Universal	75.3	11.25	0.9131 0.0049	0.7863 0.0118	0.0617
	fntbl37	MSD	155.3	13.83	0.8611 0.0062	0.7439 0.0109	
Hungarian	fntbl37	Universal	35.6	8.25	0.8093 0.0052	0.6277 0.0118	0.1753
	fntbl37	MSD	33.4	9.65	0.8077 0.0056	0.6259 0.0121	
Polish	fntbl37	Universal	13.4	8.07	0.8167 0.0046	0.6555 0.0095	0.1773
	brill24	MSD	77.5	10.17	0.7173 0.0055	0.5629 0.0078	
Romanian	fntbl37	Universal	67.4	12.69	0.8947 0.0030	0.7437 0.0070	0.0861
	brill24	MSD	52.4	12.69	0.8841 0.0034	0.7329 0.0080	
Slovak	fntbl37	Universal	53.1	8.54	0.8148 0.0050	0.6509 0.0081	0.1642
	fntbl37	MSD	112.0	10.38	0.7264 0.0061	0.5682 0.0075	
Slovenian	fntbl37	Universal	35.1	10.49	0.8579 0.0046	0.6774 0.0081	0.1282
	nltkdemo18plus	MSD	70.8	11.84	0.7899 0.0050	0.6000 0.0076	
Serbian	fntbl37	Universal	80.7	10.33	0.8397 0.0059	0.6751 0.0111	0.1307
	fntbl37	MSD	141.6	13.01	0.7525 0.0076	0.5870 0.0135	

Table 4.5: Best Results per Language for the ***NLTK***-Brill Tagger

performing better then those with a high number. But there are still differences. For example the Hungarian corpus has a similar number of out-of-vocabulary words as the Polish one (17.5 %) but the f1-score was a lot higher (0.6259).

If the simpler Universal tagset is used, results generally improve. But the differences varies from language to language. With the Hungarian corpus the f1-score increased very little from 0.6259 to 0.6277, the f1-score achieved with the Polish corpus on the other hand, increased from 0.5629 to 0.6555.
The best result of an f1-score of 0.7924 for a corpus tagged using the Universal tagset was achieved with the English corpus, the lowest one with the Hungarian corpus.

The correlation of out-of-vocabulary words and the f1-score suggest, that the Brill Tagger recognizes words, it has seen during training and is able to tag them correct. As the template set also takes tags and words into account which appear near the

	MSD Tagset			Universal Tagset		
br24	1.00			0.55		
nl18	1.00	1.00		0.55	1.00	
nl18+	1.00	1.00	1.00	0.55	1.00	1.00
	ftbl37	**br24**	**nl18**	**ftbl37**	**br24**	**nl18**

Table 4.6: Spearman Correlation for all Configurations of the nltkBrill Tagger

current word, the tagger may, to some extent, be able to recognize small sub parts of sentences, which appear frequently but we were not able to proof this in our evaluation.

The performance of the Brill Tagger is influenced by the tagset as well, as the results show. Using the Universal tagset, the results are always better then with the MSD tags. As the Universal tagset is a lot simpler then the MSD set (see section 2.5), the tagger only has to "guess" the rough type of a word, e.g. if it is a noun or a verb. With the MSD tagset on the other hand, it has to find out a lot more information. The results show that the Brill Tagger is better at guessing a rough tag based on imprecise information than finding a precise tag based on precise information.

Another thing that is influenced by the tagset is the time it takes to train the tagger. For all languages besides Hungarian and Romanian training the tagger with the *MSD* tagset took longer than training the tagger with the *universal* tagset. The difference ranges from few seconds to over double of the time used with the simpler tagset. In contrast to that, the time for tagging the test data is not effected in the same way, and also tagging is much faster than training.

In the beginning of this section we already showed that there are not many differences when using different template sets for the tagging process. The Spearman-Correlation Matrix in Table 4.6 shows that for the *MSD* tagset it is irrelevant which template set is used, all languages can be tagged equally good with each template set[18]. For the *universal* tagset the *fntbl37* template set behaves different compared to the other template sets (see Figure 4.6). In Table 4.5 it can also be seen that this template set is the best for any language with the *universal* tagset.

[18]Please note that this doesn't mean that there are no differences in the f1-score or the accuracy, this only means that if a language can be tagged best with e.g. *fntbl37* this will also be the language that could be tagged best with all other template sets.

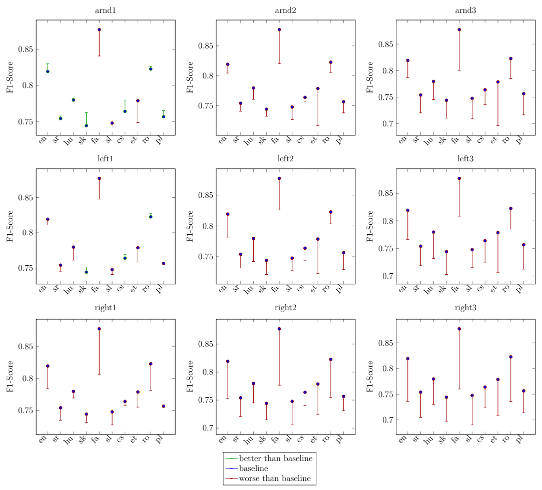

Figure 4.5: Comparison over all Languages and Configuration Options for the skMultinomialNB Tagger within the Universal Tagset

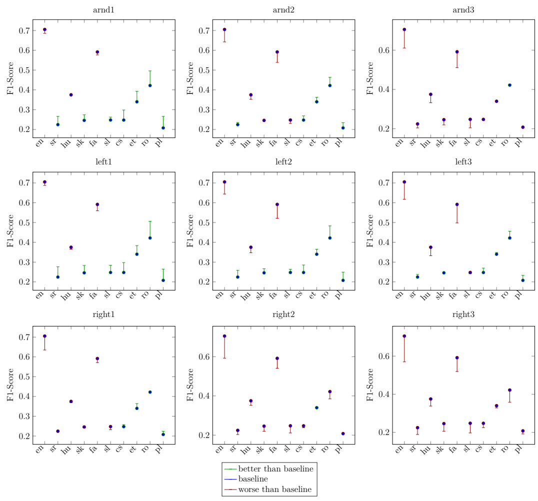

Figure 4.6: Comparison over all Languages and Configuration Options for the skMultinomialNB Tagger within the MSD Tagset

4.3.1.2 *scikit-learn*-MultinomialNB

The results of the ***scikit-learn***-MultinomialNB tagger are different to those of the Brill Tagger described in Section 4.3.1.1. When looking at Figure 4.5 one can see that with other context windows than the baseline almost no increases in performance can be measured while tagging with the *universal* tagset. For all context windows besides `arnd1` and `left1` the performance is even strictly worse than the baseline. In contrast to that, while tagging with the *MSD* tagset, all context windows except `right3` are at least able to tag one language better than with the baseline. When looking at both figures at once one can say, that overall `arnd1` and `left1` have clearly the best performance of the used context windows, independent of the used tagset.

In Table 4.7 an overview over the performance of the Brill Tagger is provided. For every language the best result according to the f1-score for the MSD and Universal tagset is shown as well as the number of out-of-vocabulary words. The name of the template set which was used to gain the results is also provided.
For every tagset the highest values are marked in green, the lowest in red.

While the language with the highest f1-score for the MSD tagset is English with a value of 0.7052 and the one with the lowest was Polish with a value of 0.2662, for the Universal tagset the outcome was quite different. Here the highest f1-score was achieved with Farsi, which has an f1-score of 0.8772, and the lowest with Slovenian, which has an f1-score of 0.7477.

Language	Configuration	Tagset	Time (seconds)		Accuracy	F1-Score	Out of Vocabulary
			Training	Tagging	(second line: standard deviation)		
Czech	[-1, 0, 1]	Universal	2.7	0.20	0.8742 0.0033	0.7796 0.0050	0.1586
	[-1, 0, 1]	MSD	11.6	0.93	0.5166 0.0060	0.2983 0.0069	
English	[-1, 0, 1]	Universal	3.2	0.21	0.9270 0.0029	0.8298 0.0071	0.0478
	[0]	MSD	2.1	0.16	0.8346 0.0037	0.7052 0.0072	
Estonian	[0]	Universal	1.0	0.09	0.8774 0.0022	0.7787 0.0043	0.1627
	[-1, 0, 1]	MSD	5.6	0.42	0.5907 0.0062	0.3933 0.0071	
Farsi	[0]	Universal	1.3	0.08	0.9327 0.0019	0.8772 0.0046	0.0617
	[0]	MSD	3.2	0.24	0.7433 0.0064	0.5916 0.0044	
Hungarian	[-1, 0, 1]	Universal	2.8	0.19	0.8736 0.0032	0.7819 0.0076	0.1753
	[0]	MSD	3.6	0.31	0.6429 0.0065	0.3750 0.0103	
Polish	[-1, 0, 1]	Universal	2.9	0.17	0.8645 0.0027	0.7649 0.0037	0.1773
	[-1, 0, 1]	MSD	9.7	0.96	0.4899 0.0072	0.2662 0.0065	
Romanian	[-1, 0]	Universal	2.4	0.14	0.9097 0.0035	0.8275 0.0081	0.0861
	[-1, 0]	MSD	5.0	0.34	0.7329 0.0026	0.5063 0.0075	
Slovak	[-1, 0, 1]	Universal	2.9	0.16	0.8573 0.0032	0.7624 0.0047	0.1642
	[-1, 0]	MSD	8.5	0.93	0.5072 0.0056	0.2827 0.0064	
Slovenian	[0]	Universal	1.2	0.12	0.8849 0.0042	0.7477 0.0073	0.1282
	[-1, 0]	MSD	8.0	0.68	0.5886 0.0053	0.2839 0.0074	
Serbian	[-1, 0, 1]	Universal	3.0	0.18	0.8651 0.0028	0.7576 0.0052	0.1307
	[-1, 0]	MSD	7.9	0.66	0.5397 0.0065	0.2765 0.0079	

Table 4.7: Best Results per Language for the ***scikit-learn***-MultinomialNB Tagger

The effect that the tagger performs better with the simpler tagset can be seen here as well but the improvement is a lot higher as with the Brill Tagger. While the highest improvement of the f1-score was 0.1 (Czech) with the Brill Tagger it is up to almost 0.5 (Polish) here. Compared to the Brill Tagger, the MultinomialNB tagger achieved higher f1-scores with the Universal tagset but lower ones for MSD tags for every language.

A remarkable fact here is that the lowest f1-score for the Universal tagset is still higher than the best for the MSD tags, while for the Brill Tagger the ranges for the two tagsets overlap.
The results suggest, as with the Brill Tagger, that in general the tagger performs better if the number of out-of-vocabulary words is low.

	MSD Tagset									Universal Tagset								
	arnd3	**arnd2**	**arnd1**	**base**	**left1**	**left2**	**left3**	**right1**	**right2**	**arnd3**	**arnd2**	**arnd1**	**base**	**left1**	**left2**	**left3**	**right1**	**right2**
arnd2	0.96									0.62								
arnd1	0.96	1.00								0.77	0.94							
base	0.89	0.87	0.87							0.87	0.79	0.82						
left1	0.94	0.93	0.93	0.98						0.89	0.83	0.90	0.93					
left2	0.96	0.96	0.96	0.95	0.99					0.62	0.95	0.87	0.84	0.84				
left3	0.96	0.96	0.96	0.95	0.99	1.00				0.62	0.94	0.83	0.84	0.79	0.98			
right1	0.98	0.94	0.94	0.95	0.96	0.98	0.98			0.84	0.89	0.95	0.94	0.95	0.89	0.85		
right2	0.98	0.94	0.94	0.95	0.96	0.98	0.98	1.00		0.82	0.90	0.94	0.95	0.96	0.90	0.87	0.99	
right3	0.98	0.94	0.94	0.95	0.96	0.98	0.98	1.00	1.00	0.82	0.90	0.94	0.95	0.96	0.90	0.87	0.99	1.00

Table 4.8: Spearman Correlation across all Configurations for the ***scikit-learn***-MultinomialNB Tagger

The configurations that yielded the best results point out, that taking more than one word before and after the current word into account reduces the performance of the tagger. Also, no configuration, that only considers words that appear after the current one, can be found amongst the best ones. But the best configuration depends on the language. While in Polish the best results were achieved using a context window of 3 words (around the current word), the tagger performed best for Farsi, when it only used the current word. With some languages the best configuration depends on the tagset as well. For Estonian a configuration with a 3-word window performed best for MSD tags while the configuration that takes only the current word into account performed best for the simpler Universal tagset. There are also languages where configurations that use a larger context window perform better for the Universal tagset and ones that use smaller ones perform better for MSD tags, like for example Hungarian.

What can also bee seen from Table 4.7 is that the training as well as tagging times are drastically lower than for the Brill Tagger. But still, training the tagger with an *MSD* tagged text takes longer than for a text tagged with the *universal* tagset.

The Spearman Rank Correlation Matrix in Table 4.8 is very homogeneous for the *MSD* tagset. This means that with the ***scikit-learn***-MultinomialNB Tagger languages are for all context windows equally hard to tag. For the *universal* tagset the matrix is not that homogeneous, but still `right2` and `right3` are equal (as well as for the *MSD* tagset, where also `right1` is equal to both others).

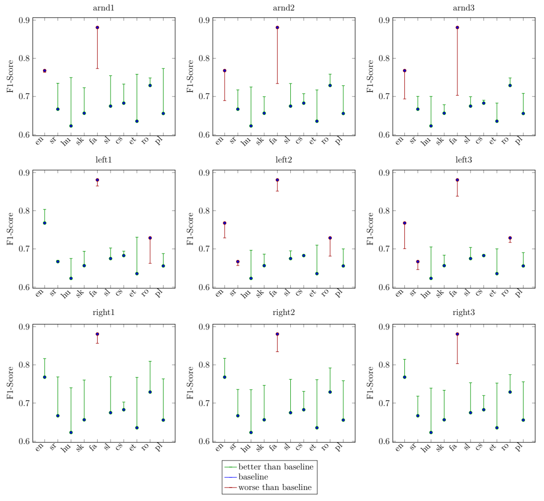

Figure 4.7: Comparison over all Languages and Configuration Options for the skPerceptron Tagger within the Universal Tagset

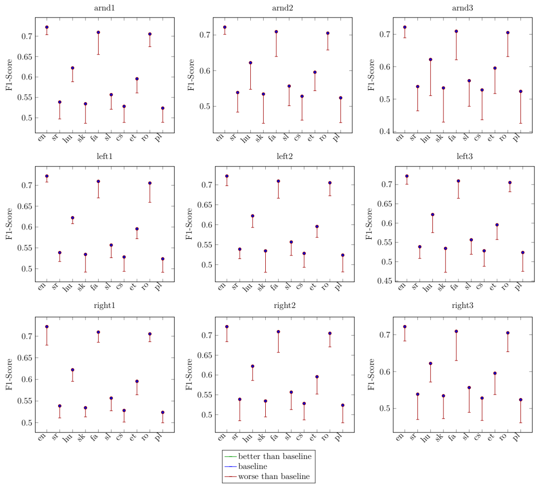

Figure 4.8: Comparison over all Languages and Configuration Options for the skPerceptron Tagger within the MSD Tagset

4.3.1.3 *scikit-learn*-Perceptron

The results of the Perceptron tagger are very similar to those of the MultinomialNB which are shown in section 4.3.1.2. But when comparing the plots in Figures 4.5, 4.6, 4.7 and 4.8 it can be seen that opposing to the `skMultinomialNB` Tagger no performance increases can be measured for the *MSD* tagset, but this time for the *universal* tagset. For the latter, the best context windows are `right1`, `right2` and `right3`, for all other context windows at least one more language could not be tagged better than the baseline. The only language which could be tagged best with the baseline is Farsi. For the *MSD* tagset no increase in performance could be measured at all, the baseline is strictly better than any other context windows.

In Table 4.9 an overview over the performance of the Brill Tagger is provided. For every language the best result according to the f1-score for the MSD and Universal tagset is shown as well as the number of out-of-vocabulary words. The name of the template set which was used to gain the results is also provided.
For every tagset the highest values are marked in green, the lowest in red.

The best f1-scores were achieved for the English language and the MSD tags with a value of 0.7220 and for Farsi and the *universal* tagset with a value of 0.8808. The lowest score, 0.5238, for the MSD tagset was achieved with Polish, only the language with the lowest score for the Universal tagset differs. Here it is Czech with a value of 0.7326.

Language	Configuration	Tagset	Time (seconds) Training	Tagging	Accuracy (second line: standard deviation)	F1-Score	Out of Vocabulary
Czech	[-1, 0, 1]	Universal	10.7	0.10	0.8521 0.0053	0.7326 0.0101	0.1586
	[0]	MSD	529.6	0.33	0.6782 0.0078	0.5282 0.0077	
English	[0, 1, 2]	Universal	15.9	0.11	0.8984 0.0040	0.8171 0.0095	0.0478
	[0]	MSD	126.5	0.07	0.8373 0.0082	0.7220 0.0081	
Estonian	[0, 1]	Universal	8.4	0.06	0.8730 0.0064	0.7672 0.0092	0.1627
	[0]	MSD	213.9	0.14	0.7277 0.0121	0.5956 0.0117	
Farsi	[0]	Universal	7.7	0.04	0.9203 0.0106	0.8808 0.0039	0.0617
	[0]	MSD	324.0	0.15	0.7881 0.0117	0.7093 0.0109	
Hungarian	[-1, 0, 1]	Universal	10.5	0.08	0.8587 0.0050	0.7495 0.0119	0.1753
	[0]	MSD	339.2	0.22	0.7968 0.0038	0.6222 0.0124	
Polish	[-1, 0, 1]	Universal	12.5	0.08	0.8738 0.0071	0.7733 0.0114	0.1773
	[0]	MSD	417.7	0.32	0.6523 0.0104	0.5238 0.0086	
Romanian	[0, 1]	Universal	14.4	0.05	0.8958 0.0044	0.8095 0.0083	0.0861
	[0]	MSD	315.2	0.16	0.8421 0.0062	0.7052 0.0092	
Slovak	[0, 1]	Universal	10.4	0.05	0.8567 0.0083	0.7603 0.0126	0.1642
	[0]	MSD	583.9	0.42	0.6759 0.0086	0.5344 0.0089	
Slovenian	[0, 1]	Universal	13.5	0.07	0.8911 0.0084	0.7687 0.0168	0.1282
	[0]	MSD	680.6	0.42	0.7384 0.0097	0.5567 0.0078	
Serbian	[0, 1]	Universal	11.2	0.09	0.8524 0.0085	0.7685 0.0112	0.1307
	[0]	MSD	650.6	0.37	0.6729 0.0169	0.5387 0.0117	

Table 4.9: Best Results per Language for the ***scikit-learn***-Perceptron Tagger

As with the MultinomialNB tagger all scores achieved with the Universal tagset are higher than those for the MSD tags. The results also suggest that the number of out-of-vocabulary words influence the score.

A remarkable fact concerning the context windows is, that all configurations which performed best with the MSD tagset only take the current word into account while, except for Farsi, with the Universal tagset the context window extended at least one word to the right.In general it seems that the tagger performs best with small context windows.

Additionally it seems that the word that follows the current one is of more importance to the Perceptron tagger than the preceding one, as all configurations that use multiple words use at least one following, but only three use the preceding word.

	MSD Tagset									Universal Tagset								
arnd2	0.98									0.16								
arnd1	0.95	0.99								0.30	0.31							
base	0.96	0.95	0.94							0.50	0.16	0.12						
left1	0.98	0.98	0.95	0.99						0.31	-0.26	0.48	0.38					
left2	0.98	1.00	0.99	0.95	0.98					0.37	-0.05	0.81	0.12	0.79				
left3	0.98	1.00	0.99	0.95	0.98	1.00				0.70	0.65	0.48	0.36	0.24	0.47			
right1	0.92	0.89	0.88	0.94	0.92	0.89	0.89			0.71	0.38	0.50	0.70	0.37	0.37	0.54		
right2	0.94	0.92	0.89	0.95	0.94	0.92	0.92	0.95		0.68	0.36	0.61	0.62	0.50	0.56	0.64	0.93	
right3	0.95	0.93	0.92	0.98	0.95	0.93	0.93	0.96	0.99	0.65	0.32	0.75	0.53	0.44	0.67	0.72	0.75	0.89
	arnd3	**arnd2**	**arnd1**	**base**	**left1**	**left2**	**left3**	**right1**	**right2**	**arnd3**	**arnd2**	**arnd1**	**base**	**left1**	**left2**	**left3**	**right1**	**right2**

Table 4.10: Spearman Correlation across all Configurations for the ***scikit-learn***-Perceptron Tagger

What also can be seen from Table 4.9 is that the training times are much higher than for the MultinomialNB Tagger. While the training times for the texts tagged with the *universal* tagset are still shorter than the training times of the corresponding texts for the Brill Tagger, they are also higher than for the MultinomialNB Tagger. When looking at the texts tagged with the *MSD* tagset exceed the times of the Brill and the MultinomialNB tagger. The language that could be tagged in the shortest time was English in 126.5 seconds (c.f. 2.1 seconds for the MultinomialNB Tagger) and the slowest training time, 680.6 seconds, was achieved with the Slovenian version of the text (c.f. 8.0 seconds for the MultinomialNB Tagger).

The Spearman Rank Correlation Matrix in Table 4.10 is very homogeneous for the *MSD* tagset, almost all values are higher than 0.9. This means that with the ***scikit-learn***-Perceptron Tagger languages are for all context windows equally hard to tag. For the *universal* tagset the matrix is not homogeneous at all, values range from -0.26 to 0.93, where the latter is the only value higher than 0.9. This shows that for the *universal* tagset the context windows is important and as one can see in Figure 4.7 the choice of the context window has a huge effect on the performance of the tagger, but also the language is important.

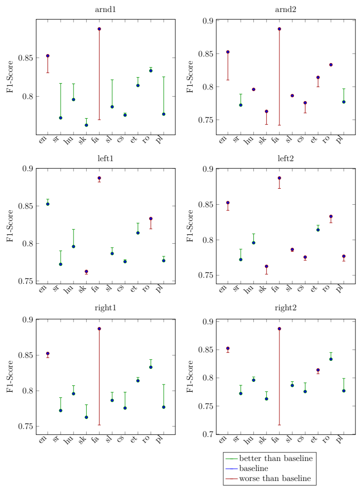

Figure 4.9: Comparison over all Languages and Configuration Options for the skLinearSVC Tagger within the Universal Tagset

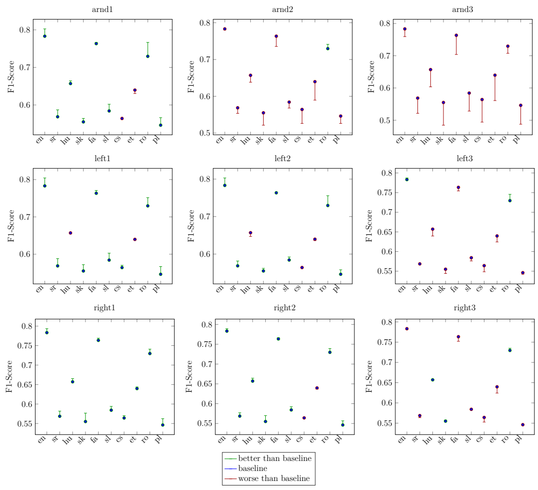

Figure 4.10: Comparison over all Languages and Configuration Options for the skLinearSVC Tagger within the MSD Tagset

4.3.1.4 *scikit-learn*-LinearSVC

Compared to the MultinomialNB and Perceptron tagger one can see from Figures 4.9 and 4.10 that the baseline is not as good as for the other two mentioned taggers, regardless of the chosen tagset. For the *universal* tagset it seems that all configurations of `skLinearSVC` containing at least one word right of the current word are for Farsi really bad. In general the `left1`, and (disregarding Farsi) `arnd1` and `right1` provide the best performance.
For the *MSD* tagset the context window `right1` is strictly better than the baseline, `arnd1` and `left1` are also good, for some languages even better than `right1` but they also have drawbacks while tagging Hungarian, Estonian and Czech.

In Table 4.9 an overview over the performance of the Brill Tagger is provided. For every language the best result according to the f1-score for the MSD and Universal tagset is shown as well as the number of out-of-vocabulary words. The name of the template set which was used to gain the results is also provided.
For every tagset the highest values are marked in green, the lowest in red.

The best f1-scores have been achieved for the MSD tagset with English, which has a value of 0.8044, and for the *universal* tagset with Farsi, which has a value of 0.8874. The lowest f1-score, 0.5668, for the MSD tagset came out for Polish. For Universal tagset not Polish but Slovak was the worst taggable language with an f1-score of 0.7804. Here the ranges of scores for the different tagsets overlap as with the Brill Tagger. Again, the scores tend to be higher if the number of out-of-vocabulary words is lower.

Language	Configuration	Tagset	Time (seconds)		Accuracy	F1-Score	Out of Vocabulary
			Training	Tagging	(second line: standard deviation)		
Czech	[0, 1]	Universal	14.6	0.05	0.8908 0.0031	0.7981 0.0055	0.1586
	[0, 1]	MSD	930.1	0.21	0.7195 0.0057	0.5703 0.0073	
English	[-1, 0]	Universal	14.7	0.09	0.9316 0.0018	0.8590 0.0067	0.0478
	[-1, 0]	MSD	187.4	0.14	0.8914 0.0038	0.8044 0.0080	
Estonian	[-1, 0]	Universal	15.4	0.11	0.9067 0.0032	0.8271 0.0050	0.1627
	[0, 1]	MSD	435.1	0.11	0.7714 0.0063	0.6439 0.0087	
Farsi	[0]	Universal	27.9	0.06	0.9376 0.0015	0.8874 0.0039	0.0617
	[-1, 0]	MSD	469.5	0.12	0.8483 0.0054	0.7709 0.0117	
Hungarian	[-1, 0]	Universal	16.7	0.17	0.8985 0.0040	0.8188 0.0093	0.1753
	[0, 1]	MSD	747.7	0.17	0.8269 0.0045	0.6652 0.0103	
Polish	[-1, 0, 1]	Universal	17.7	0.13	0.9042 0.0029	0.8252 0.0047	0.1773
	[-1, 0]	MSD	866.9	0.26	0.6991 0.0050	0.5668 0.0085	
Romanian	[0, 1, 2]	Universal	19.9	0.14	0.9179 0.0029	0.8452 0.0052	0.0861
	[-1, 0, 1]	MSD	294.6	0.18	0.8685 0.0037	0.7666 0.0079	
Slovak	[0, 1]	Universal	15.5	0.08	0.8722 0.0039	0.7804 0.0063	0.1642
	[0, 1]	MSD	987.1	0.24	0.7191 0.0073	0.5767 0.0094	
Slovenian	[-1, 0, 1]	Universal	16.3	0.13	0.9170 0.0041	0.8215 0.0087	0.1282
	[-1, 0]	MSD	1228.6	0.25	0.7801 0.0049	0.6027 0.0074	
Serbian	[-1, 0, 1]	Universal	17.1	0.09	0.8817 0.0034	0.8167 0.0058	0.1307
	[-1, 0]	MSD	1028.0	0.26	0.7343 0.0065	0.5881 0.0122	

Table 4.11: Best Results per Language for the ***scikit-learn***-LinearSVC Tagger

It seems that the LinearSVC tagger needs larger context windows to perform well, as nine of the best configurations use a window size of three words but very large window won't yield very good results, either, as non of the best configurations use more than that. It also seems that considering only the current word does not work as well with this tagger as with the others as only one of those configurations can be found amongst the best ones.
A preference to using preceding or following words can not be found, but if preceding words are used, the tagger sometimes takes two of them into account, while it uses only one following in the configurations that yield the best results.

What also can be seen from Table 4.11 is that the training times with the *MSD* tagset tagged texts are the highest of all tested configurations. They go up to over 1000 seconds. Training the tagger with texts tagged with the *universal* tagset takes approximately the same time as for the `skPerceptron` tagger but is overall still a

	MSD Tagset									Universal Tagset								
arnd2	1.00									0.49								
arnd1	0.95	0.95								0.39	0.94							
base	0.98	0.98	0.95							0.90	0.36	0.30						
left1	0.95	0.95	0.96	0.99						0.90	0.33	0.25	0.95					
left2	0.98	0.98	0.95	1.00	0.99					0.94	0.32	0.21	0.92	0.96				
left3	0.98	0.98	0.95	1.00	0.99	1.00				0.94	0.32	0.21	0.92	0.96	1.00			
right1	0.95	0.95	0.96	0.99	1.00	0.99	0.99			0.48	0.94	0.88	0.39	0.31	0.28	0.28		
right2	0.95	0.95	0.96	0.99	1.00	0.99	0.99	1.00		0.55	0.95	0.88	0.43	0.36	0.33	0.33	0.96	
right3	0.95	0.95	0.96	0.99	1.00	0.99	0.99	1.00	1.00	0.55	0.95	0.88	0.43	0.36	0.33	0.33	0.96	1.0
	arnd3	**arnd2**	**arnd1**	**base**	**left1**	**left2**	**left3**	**right1**	**right2**	**arnd3**	**arnd2**	**arnd1**	**base**	**left1**	**left2**	**left3**	**right1**	**right2**

Table 4.12: Spearman Correlation across all Configurations for the ***scikit-learn***-LinearSVC Tagger

bit higher. Just like the other ***scikit-learn***-based taggers, the tagging times are very fast and do not exceed 0.5 seconds.

The Spearman Rank Correlation Matrix in Table 4.12 is very homogeneous for the *MSD* tagset, all values are higher than or equal to 0.95[19]. This means that with the ***scikit-learn***-LinearSVC Tagger languages are for all context windows equally hard to tag. For the *universal* tagset the matrix is more diverse. The values range from 0.21 to 1.0. This shows that for the *universal* tagset the context windows is important and as one can see in Figure 4.9 the choice of the context window has a huge effect on the performance of the tagger, but also the language is important.

4.3.1.5 Conclusion for the Analysis per Tagger Evaluation

Independent of the language and the used tagger, the f1-score for *MSD* tagged texts is strictly lower than for the same text tagged with the *universal* tagset. This is a result of the more complicated tagset which has to be dealt with, and also the fact that our taggers are not specialized on feature-based tagsets. This means, that the tag has to be complete in order to be correct, where complete means that the computed tag has to be exactly the same as the one from the gold standard, thus no feature of the tag may be left out[20].
What also comes to mind when seeing that it is mostly unimportant to chose a certain configuration when tagging texts with the *MSD* tagset, is that it is too complicated for the taggers to find the relations between two or more tags because of the huge amount of different tags in contrast to the small amount of tags for the *universal* tagset, where the configuration has more impact.

[19] This reminds of the `skPerceptron` tagger where it was almost the same for the *MSD* tagset

[20] For such a task we would need to write a completely new tagger, it could not be done with the tools we used. But we think that this could be a good way to increase the performance on the *MSD* tagged texts, e.g. when for a word only it could be computed that it is a verb, but not the tense, we could tag it partly, what would be better than tagging it with wrong additional features.

	Czech	English	Estonian	Farsi	Hungarian	Polish	Romanian	Serbian	Slovak
English	0.80								
Estonian	0.97	0.83							
Farsi	0.77	0.74	0.78						
Hungarian	0.96	0.83	0.97	0.75					
Polish	0.98	0.80	0.98	0.75	0.96				
Romanian	0.92	0.86	0.91	0.67	0.94	0.92			
Serbian	0.98	0.83	0.98	0.76	0.96	0.99	0.94		
Slovak	0.99	0.83	0.98	0.78	0.96	0.99	0.92	0.99	
Slovenian	0.98	0.82	0.98	0.77	0.96	0.99	0.92	0.99	0.99

Table 4.13: Spearman Correlation across all Configurations, Taggers and Tagsets for all Languages

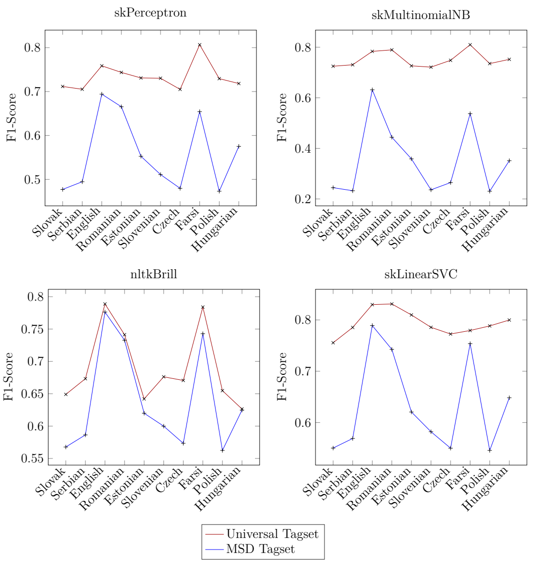

Figure 4.11: Comparison of performance between MSD and Universal Tagset

4.3.2 Analysis per Language

In contrast to the last section which was divided per tagger and the languages were compared, we have a look at the evaluation results, sorted per language in this section. This way we can compare the performance of the taggers on each language separately. In general, in Figure 4.11 one can see that for every language and tagger, the *universal* tags can be handled better than the *MSD* tags. However, the performance difference between *universal* -and *MSD*-tagged texts differs from tagger to tagger. It can also be seen that there are languages that can be tagged better and some that can be tagged worse, e.g. English and Farsi can be tagged good with either the *MSD* or the *universal* tagset for all taggers, in contrast to that, Slovak and Polish can be tagged much better with the *universal* tagset, but overall both are worse than English and Farsi.

That there are languages that can be tagged equally good or bad can also be seen from the Spearman Correlation Matrice in Table 4.13, While English and Farsi don't have much in common with other languages, Polish, Slovak, Serbian, Hungarian and Romanian have values ranging from 0.92 to 0.99 which leads to the conclusion that – with the tested configurations – the texts are equally hard to tag, and therefore the languages may have something in common.

Tagger	Configuration	Tagset	Time (seconds) Training	Tagging	Accuracy (second line: standard deviation)	F1-Score	Out of Vocabulary
skLinearSVC	[0, 1]	Universal	14.6	0.05	0.8908 0.0031	0.7981 0.0055	0.1586
skMultinomialNB	[-1, 0, 1]	Universal	2.7	0.20	0.8742 0.0033	0.7796 0.0050	
skPerceptron	[-1, 0, 1]	Universal	10.7	0.10	0.8521 0.0053	0.7326 0.0101	
nltkBrill	fntbl37	Universal	26.2	7.82	0.8296 0.0047	0.6716 0.0075	
nltkBrill	brill24	MSD	79.5	10.53	0.7324 0.0057	0.5737 0.0062	
skLinearSVC	[0, 1]	MSD	930.1	0.21	0.7195 0.0057	0.5703 0.0073	
skPerceptron	[0]	MSD	529.6	0.33	0.6782 0.0078	0.5282 0.0077	
skMultinomialNB	[-1, 0, 1]	MSD	11.6	0.93	0.5166 0.0060	0.2983 0.0069	

Table 4.14: Best Results per Tagger for the Czech Language

4.3.2.1 Czech

In table 4.14 the best results of each tagger for the Czech language and each tagset can be found.
For the Czech language it seems that it is very hard to tag text with the MSD tags as compared to the simpler Universal tagset as the best f1-score for MSD tags (0.5737), achieved by the Brill Tagger, is by almost 0.1 lower than the lowest one for Universal tags (0.6716, by the Brill Tagger as well).
The LinearSVC tagger, achieved the highest f1-score for the Universal tagset (0.7981) and around 0.22 less with the MSD tags (0.5703) but is still almost as good as the Brill Tagger. The lowest loss of score from Universal to MSD tags has the Brill tagger with a loss of about 0.1, the highest loss has the MultinomialNB, it looses about 0.48 and achieved the lowest score for MSD tags (0.2983).
The time consumed for training increases for each tagger from Universal to MSD tags. But while the Brill Tagger needs three times as long to be trained with MSD tags as it needs with Universal tags, the LinearSVC's training time increase with a factor of 66. Compared to the other taggers, the Brill Tagger needs a lot of time for tagging (7 to 10 seconds). Tagging times increase when using the MSD tags as well, but not as much as the training time.
The best configurations for LinearSVC and MultinomialNB are the same for both tagsets while it changes for the Brill and the Perceptron Tagger.

Tagger	Configuration	Tagset	Time (seconds)		Accuracy	F1-Score	Out of Vocabulary
			Training	Tagging	(second line: standard deviation)		
skLinearSVC	[-1, 0]	Universal	14.7	0.09	0.9316 0.0018	0.8590 0.0067	0.0478
skMultinomialNB	[-1, 0, 1]	Universal	3.2	0.21	0.9270 0.0029	0.8298 0.0071	
skPerceptron	[0, 1, 2]	Universal	15.9	0.11	0.8984 0.0040	0.8171 0.0095	
nltkBrill	fntbl37	Universal	105.7	15.35	0.9263 0.0023	0.7924 0.0080	
skLinearSVC	[-1, 0]	MSD	187.4	0.14	0.8914 0.0038	0.8044 0.0080	
nltkBrill	fntbl37	MSD	124.4	14.16	0.9087 0.0024	0.7792 0.0065	
skPerceptron	[0]	MSD	126.5	0.07	0.8373 0.0082	0.7220 0.0081	
skMultinomialNB	[0]	MSD	2.1	0.16	0.8346 0.0037	0.7052 0.0072	

Table 4.15: Best Results per Tagger for the English Language

4.3.2.2 English

In table 4.15 the best results of each tagger for the English language and each tagset can be found.
The results for English with the Universal tagset are quite good for all taggers. The best f1-score was achieved by the LinearSVC tagger (0.8590), the lowest by the Brill Tagger was only about 0.06 less (0.7924).
With the MSD tags all taggers achieved a lower score. The LinearSVC achieved the highest score (0.8044) while the MultinomialNB tagger got the lowest (0.7052). The smallest difference has the Brill tagger with only a little bit over 0.01 while the highest difference (about 0.12) occurs with the MultinomialNB.
For all taggers, except for the MultinomialNB, training time is higher for Universal tags than for MSD tags. While the time hardly increases with the Brill Tagger, the LinearSVC tagger needs 13 times as much time to be trained with MSD tags than with the Universal tagset.
Tagging time is lower for MSD tags than for Universal tags with all taggers, except the LinearSVC.
The best configurations for the LinearSVC and Brill Tagger are identical for both tagsets, while they differ for the Perceptron and MultinomialNB taggers.

Tagger	Configuration	Tagset	Time (seconds) Training	Tagging	Accuracy (second line: standard deviation)	F1-Score	Out of Vocabulary
skLinearSVC	[-1, 0]	Universal	15.4	0.11	0.9067 0.0032	0.8271 0.0050	0.1627
skMultinomialNB	[0]	Universal	1.0	0.09	0.8774 0.0022	0.7787 0.0043	
skPerceptron	[0, 1]	Universal	8.4	0.06	0.8730 0.0064	0.7672 0.0092	
nltkBrill	fntbl37	Universal	32.4	7.57	0.8209 0.0068	0.6434 0.0109	
skLinearSVC	[0, 1]	MSD	435.1	0.11	0.7714 0.0063	0.6439 0.0087	
nltkBrill	fntbl37	MSD	69.9	8.74	0.7793 0.0066	0.6207 0.0104	
skPerceptron	[0]	MSD	213.9	0.14	0.7277 0.0121	0.5956 0.0117	
skMultinomialNB	[-1, 0, 1]	MSD	5.6	0.42	0.5907 0.0062	0.3933 0.0071	

Table 4.16: Best Results per Tagger for the Estonian Language

4.3.2.3 Estonian

In table 4.16 the best results of each tagger for the Estonian language and each tagset can be found.
With the Universal tagset the best f1-score was achieved by the LinearSVC tagger (0.8271), the lowest by the Brill Tagger (0.6434), the difference is about 0.18.
The best score for the MSD tags (0.6439 by LinearSVC) is hardly better then the lowest for the Universal tags. The lowest score for the MSD tagset (0.3933) was achieved by the MultinomialNB tagger. The difference between the highest ant lowest score is about 0.25.
The training and tagging times with the MSD tags are higher than those for the Universal tagset with all taggers, except for the LinearSVC where tagging takes exactly the same time for both tagsets.
Best configurations for each tagset per tagger differ for all taggers except for the Brill Tagger where the *fntbl37* template set performs best for both tagsets.

Tagger	Configuration	Tagset	Time (seconds)		Accuracy	F1-Score	Out of Vocabulary
			Training	Tagging	(second line: standard deviation)		
skLinearSVC	[0]	Universal	27.9	0.06	0.9376 0.0015	0.8874 0.0039	0.0617
skPerceptron	[0]	Universal	7.7	0.04	0.9203 0.0106	0.8808 0.0039	
skMultinomialNB	[0]	Universal	1.3	0.08	0.9327 0.0019	0.8772 0.0046	
nltkBrill	fntbl37	Universal	75.3	11.25	0.9131 0.0049	0.7863 0.0118	
skLinearSVC	[-1, 0]	MSD	469.5	0.12	0.8483 0.0054	0.7709 0.0117	
nltkBrill	fntbl37	MSD	155.3	13.83	0.8611 0.0062	0.7439 0.0109	
skPerceptron	[0]	MSD	324.0	0.15	0.7881 0.0117	0.7093 0.0109	
skMultinomialNB	[0]	MSD	3.2	0.24	0.7433 0.0064	0.5916 0.0044	

Table 4.17: Best Results per Tagger for the Farsi Language

4.3.2.4 Farsi

In table 4.17 the best results of each tagger for Farsi and each tagset can be found. With the Universal tagset the highest score (0.8874) was achieved by the LinearSVC tagger, the lowest (0.7863) by the Brill tagger.
In this language all results for MSD tags were lower than the lowest one for the Universal tags. The highest score was achieved by the LinearSVC (0.7709), the lowest by the MultinomialNB (0.5916).
The Brill Tagger's score was similar in both tagsets, the difference is about 0.04.
For each tagger the training and tagging times are higher for the MSD tagset than for the Universal tagset.
It is remarkable that for all taggers that use context windows, the best configuration is to take only the current word into account, except for the LinearSVC tagger with the MSD tagset, where the best option is to use the preceding word as well. For the Brill Tagger the best performing template set is *fntbl37* for both tagsets.

Tagger	Configuration	Tagset	Time (seconds)		Accuracy	F1-Score	Out of Vocabulary
			Training	Tagging	(second line: standard deviation)		
skLinearSVC	[-1, 0]	Universal	16.7	0.17	0.8985 0.0040	0.8188 0.0093	0.1753
skMultinomialNB	[-1, 0, 1]	Universal	2.8	0.19	0.8736 0.0032	0.7819 0.0076	
skPerceptron	[-1, 0, 1]	Universal	10.5	0.08	0.8587 0.0050	0.7495 0.0119	
nltkBrill	fntbl37	Universal	35.6	8.25	0.8093 0.0052	0.6277 0.0118	
skLinearSVC	[0, 1]	MSD	747.7	0.17	0.8269 0.0045	0.6652 0.0103	
nltkBrill	fntbl37	MSD	33.4	9.65	0.8077 0.0056	0.6259 0.0121	
skPerceptron	[0]	MSD	339.2	0.22	0.7968 0.0038	0.6222 0.0124	
skMultinomialNB	[0]	MSD	3.6	0.31	0.6429 0.0065	0.3750 0.0103	

Table 4.18: Best Results per Tagger for the Hungarian Language

4.3.2.5 Hungarian

In table 4.18 the best results of each tagger for the Hungarian language and each tagset can be found.
The best f1-score for the Universal tagset (0.8188) was achieved by the LinearSVC tagger, the lowest one (0.6277) by the Brill Tagger.
Similar to the other languages, the LinearSVC tagger perfored best for the MSD tagset (0.6652) while the MultinomialNB tagger achieved the lowest score (0.3750). Here as well, the Brill Tagger performs almost as good for the MSD tagset as for the Universal tags.
The Brill Tagger's training time for MSD tags is lower than that for the Universal tagset. For the other taggers it is the other way around. Tagging times are higher or equal for the MSD tagset than for Universal tags.
Best performing configurations are different for each tagger except for the Brill Tagger, where the *fntbl37* templates perform best for both tagsets.

Tagger	Configuration	Tagset	Time (seconds) Training	Tagging	Accuracy (second line: standard deviation)	F1-Score	Out of Vocabulary
skLinearSVC	[-1, 0, 1]	Universal	17.7	0.13	0.9042 0.0029	0.8252 0.0047	0.1773
skPerceptron	[-1, 0, 1]	Universal	12.5	0.08	0.8738 0.0071	0.7733 0.0114	
skMultinomialNB	[-1, 0, 1]	Universal	2.9	0.17	0.8645 0.0027	0.7649 0.0037	
nltkBrill	fntbl37	Universal	13.4	8.07	0.8167 0.0046	0.6555 0.0095	
skLinearSVC	[-1, 0]	MSD	866.9	0.26	0.6991 0.0050	0.5668 0.0085	
nltkBrill	brill24	MSD	77.5	10.17	0.7173 0.0055	0.5629 0.0078	
skPerceptron	[0]	MSD	417.7	0.32	0.6523 0.0104	0.5238 0.0086	
skMultinomialNB	[-1, 0, 1]	MSD	9.7	0.96	0.4899 0.0072	0.2662 0.0065	

Table 4.19: Best Results per Tagger for the Polish Language

4.3.2.6 Polish

In table 4.19 the best results of each tagger for the Polish language and each tagset can be found.
With Polish the taggers performed a lot worse for MSD tags than for Universal tags. The best score for MSD tags (0.5668 by LinearSVC) is about 0.09 lower than the lowest for the Universal tagset (0.6555 by Brill Tagger).
The LinearSVC achieved the highest score for Universal tags (0.8252), the MultinomialNB the lowest for MSD tags (0.2662). For MSD tags, LinearSVC and Brill tagger performed almost equally well.
For Polish all taggers needed more time for both, training and tagging, for the MSD tagset than for Universal tags.
Except for the MultinomialNB, where the same configuration performed best for both tagsets, the best configuration changed for all taggers.

Tagger	Configuration	Tagset	Time (seconds)		Accuracy	F1-Score	Out of Vocabulary
			Training	Tagging	(second line: standard deviation)		
skLinearSVC	[0, 1, 2]	Universal	19.9	0.14	0.9179 0.0029	0.8452 0.0052	0.0861
skMultinomialNB	[-1, 0]	Universal	2.4	0.14	0.9097 0.0035	0.8275 0.0081	
skPerceptron	[0, 1]	Universal	14.4	0.05	0.8958 0.0044	0.8095 0.0083	
nltkBrill	fntbl37	Universal	67.4	12.69	0.8947 0.0030	0.7437 0.0070	
skLinearSVC	[-1, 0, 1]	MSD	294.6	0.18	0.8685 0.0037	0.7666 0.0079	
nltkBrill	brill24	MSD	52.4	12.69	0.8841 0.0034	0.7329 0.0080	
skPerceptron	[0]	MSD	315.2	0.16	0.8421 0.0062	0.7052 0.0092	
skMultinomialNB	[-1, 0]	MSD	5.0	0.34	0.7329 0.0026	0.5063 0.0075	

Table 4.20: Best Results per Tagger for the Romanian Language

4.3.2.7 Romanian

In table 4.20 the best results of each tagger for the Romanian language and each tagset can be found.
For the Romanian language the results were quite good compared to some other languages. This might be because of the low amount of out-of-vocabulary words.
The highest f1-scores (Universal: 0.8452; MSD: 0.7666) were achieved by the LinearSVC tagger. The lowest score for Universal tagset (0.7437) came out with the Brill Tagger, for the Universal tagset it was the MultinomialNB (0.5063). The Brill Tagger needed less time to be trained for the MSD tagset than for Universal tags and an equal amount for tagging for both tagsets. Except for that, all training and tagging times were higher for MSD tags than for Universal tags.
The best performing configuration was the same for both tagsets with the MultinomialNB tagger but a different one for all other taggers and tagsets.

Tagger	Configuration	Tagset	Time (seconds)		Accuracy	F1-Score	Out of Vocabulary
			Training	Tagging	(second line: standard deviation)		
skLinearSVC	[-1, 0, 1]	Universal	17.1	0.09	0.8817 0.0034	0.8167 0.0058	0.1307
skPerceptron	[0, 1]	Universal	11.2	0.09	0.8524 0.0085	0.7685 0.0112	
skMultinomialNB	[-1, 0, 1]	Universal	3.0	0.18	0.8651 0.0028	0.7576 0.0052	
nltkBrill	fntbl37	Universal	80.7	10.33	0.8397 0.0059	0.6751 0.0111	
skLinearSVC	[-1, 0]	MSD	1028.0	0.26	0.7343 0.0065	0.5881 0.0122	
nltkBrill	fntbl37	MSD	141.6	13.01	0.7525 0.0076	0.5870 0.0135	
skPerceptron	[0]	MSD	650.6	0.37	0.6729 0.0169	0.5387 0.0117	
skMultinomialNB	[-1, 0]	MSD	7.9	0.66	0.5397 0.0065	0.2765 0.0079	

Table 4.21: Best Results per Tagger for the Serbian Language

4.3.2.8 Serbian

In table 4.21 the best results of each tagger for Serbian and each tagset can be found.
Serbian is one of those languages where the lowest score for the Universal tagset (0.6751 by the Brill Tagger) was still higher than the lowest for MSD tags (0.5881 by the LinearSVC).
The highest score for the Universal tagset (0.8167) was achieved by the LinearSVC as well. The lowest score for MSD tags (0.2765) was achieved by the MultinomialNB tagger.
Here as well, all taggers used more time for training and tagging for the MSD tagset than for Universal tags. For the LinearSVC tagger the training time for MSD tags was exceptionally high with over a thousand seconds.

Tagger	Configuration	Tagset	Time (seconds)		Accuracy	F1-Score	Out of Vocabulary
			Training	Tagging	(second line: standard deviation)		
skLinearSVC	[0, 1]	Universal	15.5	0.08	0.8722 0.0039	0.7804 0.0063	0.1642
skMultinomialNB	[-1, 0, 1]	Universal	2.9	0.16	0.8573 0.0032	0.7624 0.0047	
skPerceptron	[0, 1]	Universal	10.4	0.05	0.8567 0.0083	0.7603 0.0126	
nltkBrill	fntbl37	Universal	53.1	8.54	0.8148 0.0050	0.6509 0.0081	
skLinearSVC	[0, 1]	MSD	987.1	0.24	0.7191 0.0073	0.5767 0.0094	
nltkBrill	fntbl37	MSD	112.0	10.38	0.7264 0.0061	0.5682 0.0075	
skPerceptron	[0]	MSD	583.9	0.42	0.6759 0.0086	0.5344 0.0089	
skMultinomialNB	[-1, 0]	MSD	8.5	0.93	0.5072 0.0056	0.2827 0.0064	

Table 4.22: Best Results per Tagger for the Slovak Language

4.3.2.9 Slovak

In table 4.22 the best results of each tagger for the Slovak language and each tagset can be found.
For Slovak, as for Serbian, the best score for MSD tags (0.5767 by the LinearSVC) was lower than the lowest for Universal tags (0.6509 by the Brill Tagger).
The highest score for Universal tags (0.7804) was achieved by the LinearSVC, the lowest for MSD tags by the MultinomialNB (0.2827). For Slovak all taggers needed more time for training and tagging with the MSD tagset than with the Universal tagset. The LinearSVC needed an exceptionally high amount of time (987.1s) for training with the MSD tags.
The LinearSVC and Brill Tagger performed best for both tagsets using the same configurations. Perceptron and MultinomialNB taggers best configurations were different for each tagset.

Tagger	Configuration	Tagset	Time (seconds) Training	Time (seconds) Tagging	Accuracy (second line: standard deviation)	F1-Score (second line: standard deviation)	Out of Vocabulary
skLinearSVC	[-1, 0, 1]	Universal	16.3	0.13	0.9170 0.0041	0.8215 0.0087	0.1282
skPerceptron	[0, 1]	Universal	13.5	0.07	0.8911 0.0084	0.7687 0.0168	
skMultinomialNB	[0]	Universal	1.2	0.12	0.8849 0.0042	0.7477 0.0073	
nltkBrill	fntbl37	Universal	35.1	10.49	0.8579 0.0046	0.6774 0.0081	
skLinearSVC	[-1, 0]	MSD	1228.6	0.25	0.7801 0.0049	0.6027 0.0074	
nltkBrill	nltkdemo18plus	MSD	70.8	11.84	0.7899 0.0050	0.6000 0.0076	
skPerceptron	[0]	MSD	680.6	0.42	0.7384 0.0097	0.5567 0.0078	
skMultinomialNB	[-1, 0]	MSD	8.0	0.68	0.5886 0.0053	0.2839 0.0074	

Table 4.23: Best Results per Tagger for the Slovenian Language

4.3.2.10 Slovenian

In table 4.23 the best results of each tagger for the Slovenian language and each tagset can be found.
As with Slovak and Serbian all results for the Universal tagset are better than the best for MSD tags. The top scores were achieved by the LinearSVC tagger (Universal: 0.8215, MSD: 0.6027), the lowest scores by the Brill Tagger (0.6774 for Universal tags) and the MultinomialNB (0.2839 for MSD tags).
All taggers needed more time for training and tagging with the MSD tagset than with Universal tags. The LinearSVC used an exceptional high amount for training with the MSD tagset (1228.6s).
Here the Brill Tagger achieved the best score for MSD tags with the *nltkdemo18plus* template set which is remarkable as it is the only language where the Brill Tagger does not achieve its best score with the *fntbl37* or the *brill24* template set. No tagger performed best with the same configuration for both tagsets.

4.3.2.11 Conclusion for the Analysis per Language Evaluation

For all languages besides Czech, using the `skLinearSVC` tagger leads to the best results for the *Universal* and the *MSD* tagset. For Czech and the *MSD* tagset the `nltkBrill` tagger is best. This tagger is for all other languages the second best tagger while tagging files with *MSD* tags. For the easier *Universal* tagset every configuration of the `nltkBrill` tagger is worse than the best configuration of any other evaluated tagger. Apart from the tagger-wise performance comparison one can see that in general English and Farsi can be tagged best, with both tagsets.

While this is true for all evaluated taggers for the *MSD* tagset, with the *Universal* tagset the differences are not that high and for the `skLinearSVC` tagger English and Romanian are the best taggable languages whereas Farsi is one of the worst.
When looking at the f1-score differences between the tagsets it is noticeable that the worst tagger[21] using the *Universal* tagset is equally good, or better than the best tagger using the *MSD* tagset.

4.3.3 Raw Data

The raw data is csv file which contains the results of every evaluation run. The first row is the tagger which produced the result, the second one indicates the used corpus, the third one the tagset, the fourth one represents the configuration option, the fifth one is the evaluated metric, followed by minimum, maximum and averaged value, the last row is the standard deviation.

For generation of the tables and graphs, we have implemented a custom toolset. It self and its documentation can be found within the `csvtools` directory in our project repository. It is important to normalize the configuration options prior to use the tools since the framework indicates which context window generator was used. In the non processed results file, `SConWin([WINDOW])` indicates a suffix based context window, whereas PConWin([WINDOW]) marks the context window as word based. For evaluating the results with our toolkit we stripped[22] those indicators and took care by ourself for the configuration options.

Taggers are named by the evaluation class of the framework. `skPerceptron` indicates that `sklearn.linear_model.Perceptron` with `n_iter=50` was uses. `skMultinomialNB` uses `sklearn.naive_bayes.MultinomialNB` with the default configuration. `skLinearSVC` instantiates `sklearn.svm.LinearSVC` with the standard settings for tagging. `nltkBrill` shows that our implementation of the ***NLTK***-brill tagger was used. `tmpOOV` is a virtual evaluation to calculate the out-of-vocabulary words.

Name of the Language can be derived from filename of the ***MULTEXT-East***-distribution. The files are named as the following scheme: oana-*ISOCODE*.xml

Tagset can either be `mte` for the *MSD* tagset or `universal` for the *universal* tagset.

[21]Worst means the ordering of the taggers only, for one tagger always its best configuration was taken for the tables.
[22]This can be done by a simple search-and-replace-command.

Configuration Option indicates the configuration for the tagger which was used. Since each implementation can define its own options we give a short overview about the ***scikit-learn*** taggers (tagger-name starts with *sk*), the ***NLTK***-brill-tagger and our helper implementation.

***scikit-learn*-Taggers** are configured by a tuple. The first value indicates the context window, which is relative to the current word (e.g. `[0,1]` indicates that the current word (*0*) and the word after it (*1*) is used.)

***NLTK*-Brill-Tagger** can use four tables for evaluation

- **fntbl37**
- **brill24**
- **nltkdemo18**
- **nltkdemo18plus**
- **baseline**: This represents the baseline and is generated by the NTLK Unigram Tagger

tmpOOV provides only the `mkOOV` option which calculates the out-of-vocabulary words for the given fold.

Metrics are calculated for each fold and summed up. For each combination we gather the following metrics: `accuracy`, `recall`, `precision`, `f1 oov`, `training_time`, `prediction_time`. If implementation does not support the metric an default value of -1 will be returned.

Minimum Value of all results from the fold is stored in this row.

Maximum Value of all results form the fold is stored in this row.

Averaged Value of all results form the fold is stored in this row.

Standard Deviation of the averaged value is stored in this row.

5 Restrictions and Challenges

In this section we list and explain all difficulties we encountered during the work on this thesis.

5.1 Python 2, encoding issues with *NLTK*

As ***MULTEXT-East*** contains textual resources in languages that are not displayable with ASCII characters, Python 2, and especially ***NLTK*** for Python 2 lead to issues while displaying things in the interactive python prompt, but also while using the `MTEBrillTagger`. Therefore we decided to write our code such that it is compatible with Python 2 and 3 and for all evaluation related things changed to the Python 3 version of ***NLTK*** where we had no problems with the encoding.

5.2 Incomplete and incomparable tags in some *MULTEXT-East* resources

During our studies we found that there are two language resources in ***MULTEXT-East*** that are different in the XML layout, `oana-bg.xml` and `oana-mk.xml`. In the Bulgarian version, instead of the `ana` attribute (which contains the MSD tags usually) the `function` attribute is used for storing the POS-Tag. However these tags are now MSD tags but have another format called ctag. These tags were used for earlier version of ***MULTEXT-East*** and have a mapping to MSD tags in the file `msd2ctag.tbl`. The mapping can only be applied when transforming the MSD tags to the ctags, the other direction does not work as the mapping is not complete. Additionally the file containing the mapping is also not complete for all languages, many of them are missing completely and some are outdated. Therefore we had to exclude the Bulgarian version of 1984 from the evaluation as it is not possible to get comparable resources with different tagsets.

The second file we had to omit in the evaluation is `oana-mk.xml` which contains the Macedonian version of 1984. It has a different XML layout and also no specific MSD tag per word, but more than one tag, where it is not sure which one should be the correct one.

5.3 Conversion of the MSD Tagset to the Penn Treebank II Tagset

In the evaluation we planned to compare the results of our POS-Tagger implementation with three tagsets: the *Universal Tagset*, the *Penn Treebank II Tagset* and the *MSD Tagset*. The reason for this was to have a rather small and abstract tagset (Universal), a more precise tagset (Penn Treebank II) and an extremely precise tagset (MSD). The conversion of MSD tags to universal tags works fine, whereas the conversion of MSD tags to Penn Treebank II tags was not possible within this work. At first it seemed doable but the huge amount of different features in an MSD Tag and the fact that there are sometimes no appropriate tags in the Penn Treebank II tagset made it impossible, moreover the Penn Treebank II tagset was made especially for English, such that it may not be possible to map some linguistic features of other languages to it. Already for the English language we encountered several problems:
For example in the Penn Treebank II tagset there is a specific tag for the word *to* named `TO`, the MSD tagset does not have this tag, we could now assume that each word tagged `#Sp` is *to* but we are not sure if this is really the case. Another example is the word *something*: In the MSD tagset it is `#Pg3ns`, a general singular pronoun in third person with a neutral gender, and in the Penn Treebank II set it is `NN` a proper noun[23][24]. So for us it was not possible to decide which tag we should use for this word, as well as for many other words. For this task much more linguistic expertise would be necessary.

[23] We found this out by using the treebank corpus of ***NLTK*** and searched for the word *something* in it.

[24] As nouns and pronouns are somehow related we checked also if there is special tags for pronouns in the Penn Treebank II tagset, but there are four different ones (`PRP, PRP$, WP, WP$`), so this did also not help deciding what should be the correct tag.

6 Conclusion

Overall, in this work we developed a corpus reader for the ***MULTEXT-East*** corpus inside ***NLTK*** and two different kinds of POS-Taggers, where one is based on the implementation of the Brill-Tagger in ***NLTK*** and the others are based on algorithms from ***scikit-learn***. The integration of ***MULTEXT-East*** into ***NLTK*** is a big improvement, as before mostly unilingual or multilingual but untagged corpora were integrated. This leads to new possibilities while comparing the taggability of languages or the performance differences of one tagger on different languages or tagsets.

To answer the question stated in the motivation, *"Is the sentence structure important for the POS-Tagger?"* we evaluated the ***MULTEXT-East***corpus with the POS-Taggers implemented by us. At the first sight the answer is yes. This can be seen by looking at the results for the different configurations of the ***scikit-learn***-based taggers. For each language there is a specific context window that produces the best results. But, the used tagset is also important. The results show that while using different tagset, different context windows provide the best results. Besides the tagset, the configuration of the tagger also has to fit to the grammar of the language of the tagged text. So the complete answer to the question is, *Yes, but besides that the used context window and the tagset have to be considered, too.*.

Our evaluation also showed, that in general the f1-score will be higher while tagging texts with the (rather simple) *Universal* tagset, compared to texts that use the *MSD* tagset, but this might also be caused by the taggers we implemented. None of them is aware of the features in an *MSD* tag, and simply takes the whole tag as one[25]. For the future it would be the best to implement a POS-Tagger specifically for the *MSD* tagset, which is aware of the different features in the tags. This way it would be possible to not only let some algorithms choose the tag that fits best to a word, but when there are more tags that would fit equally good, they could be merged and the information that is differing could just be erased from the newly assigned tag. The result would be texts that are tagged as precise as possible, however not all features that are applicable for a word would be set.

[25] As *MSD* tags are very fine-grained this leads to much more possibilities how a word can be tagged than for the *Universal* tagset.

7 Appendix

7.1 Complete Tables of Evaluation

Tagger	Configuration	Time (seconds) Training	Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	124.4	14.16	0.9087	0.7655	0.7934	0.7792
				0.0024	0.0059	0.0073	0.0065
nltkBrill	brill24	119.1	15.28	0.9067	0.7638	0.7904	0.7769
				0.0026	0.0061	0.0080	0.0068
nltkBrill	nltkdemo18	90.3	13.59	0.9039	0.7608	0.7877	0.7740
				0.0025	0.0059	0.0079	0.0067
nltkBrill	nltkdemo18plus	97.2	13.91	0.9045	0.7620	0.7888	0.7752
				0.0024	0.0059	0.0079	0.0067
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	8.4	0.44	0.7699	0.6665	0.5634	0.6106
				0.0043	0.0095	0.0071	0.0081
skMultinomialNB	[-2, -1, 0, 1, 2]	6.6	0.39	0.8015	0.6890	0.6013	0.6422
				0.0046	0.0100	0.0083	0.0089
skMultinomialNB	[-1, 0, 1]	4.4	0.21	0.8378	0.7128	0.6603	0.6855
				0.0039	0.0069	0.0065	0.0066
skMultinomialNB	[0]	2.1	0.16	0.8346	0.6750	0.7382	0.7052
				0.0037	0.0075	0.0070	0.0072
skMultinomialNB	[-1, 0]	3.3	0.15	0.8268	0.7008	0.6743	0.6873
				0.0041	0.0071	0.0072	0.0069
skMultinomialNB	[-2, -1, 0]	4.4	0.23	0.7999	0.6743	0.6161	0.6439
				0.0036	0.0075	0.0057	0.0063
skMultinomialNB	[-3, -2, -1, 0]	5.3	0.29	0.7764	0.6557	0.5830	0.6172
				0.0047	0.0089	0.0072	0.0077
skMultinomialNB	[0, 1]	3.3	0.23	0.8184	0.6475	0.6224	0.6347
				0.0043	0.0082	0.0075	0.0078
skMultinomialNB	[0, 1, 2]	4.0	0.26	0.7901	0.6229	0.5649	0.5925
				0.0051	0.0104	0.0089	0.0095
skMultinomialNB	[0, 1, 2, 3]	5.1	0.36	0.7688	0.6098	0.5348	0.5698
				0.0049	0.0092	0.0085	0.0087
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	252.9	0.32	0.8530	0.6998	0.6784	0.6889
				0.0044	0.0111	0.0122	0.0115
skPerceptron	[-2, -1, 0, 1, 2]	222.7	0.22	0.8558	0.7114	0.6922	0.7017
				0.0050	0.0101	0.0115	0.0107
skPerceptron	[-1, 0, 1]	183.2	0.14	0.8488	0.7124	0.6946	0.7034
				0.0040	0.0100	0.0087	0.0092
skPerceptron	[0]	126.5	0.07	0.8373	0.6910	0.7558	0.7220
				0.0082	0.0081	0.0084	0.0081
skPerceptron	[-1, 0]	164.3	0.09	0.8427	0.7143	0.7012	0.7077
				0.0060	0.0121	0.0148	0.0134
skPerceptron	[-2, -1, 0]	191.9	0.12	0.8410	0.7118	0.6842	0.6977
				0.0027	0.0070	0.0093	0.0080
skPerceptron	[-3, -2, -1, 0]	209.6	0.15	0.8465	0.7123	0.6898	0.7009
				0.0024	0.0085	0.0097	0.0088
skPerceptron	[0, 1]	168.5	0.10	0.8144	0.6884	0.6703	0.6792
				0.0089	0.0159	0.0181	0.0169
skPerceptron	[0, 1, 2]	198.4	0.13	0.8202	0.6991	0.6696	0.6840
				0.0026	0.0061	0.0069	0.0064
skPerceptron	[0, 1, 2, 3]	216.5	0.15	0.8258	0.6980	0.6687	0.6830
				0.0046	0.0061	0.0087	0.0071
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	95.8	0.29	0.8744	0.7585	0.7604	0.7594
				0.0032	0.0081	0.0097	0.0088
skLinearSVC	[-2, -1, 0, 1, 2]	88.8	0.21	0.8813	0.7771	0.7841	0.7806
				0.0030	0.0089	0.0095	0.0091
skLinearSVC	[-1, 0, 1]	111.0	0.14	0.8880	0.7940	0.8115	0.8026
				0.0023	0.0059	0.0070	0.0063
skLinearSVC	[0]	330.6	0.06	0.8778	0.7497	0.8199	0.7832
				0.0038	0.0070	0.0066	0.0067
skLinearSVC	[-1, 0]	187.4	0.14	0.8914	0.7883	0.8212	0.8044
				0.0038	0.0088	0.0074	0.0080
skLinearSVC	[-2, -1, 0]	145.7	0.13	0.8922	0.7915	0.8148	0.8030
				0.0037	0.0081	0.0077	0.0078
skLinearSVC	[-3, -2, -1, 0]	138.1	0.25	0.8866	0.7785	0.7962	0.7873
				0.0040	0.0095	0.0110	0.0102
skLinearSVC	[0, 1]	173.5	0.11	0.8708	0.7728	0.8150	0.7933
				0.0066	0.0068	0.0060	0.0063
skLinearSVC	[0, 1, 2]	138.8	0.17	0.8610	0.7721	0.8070	0.7892
				0.0035	0.0064	0.0065	0.0063
skLinearSVC	[0, 1, 2, 3]	115.5	0.21	0.8588	0.7666	0.7961	0.7811
				0.0036	0.0068	0.0070	0.0068

Table 7.1: Results of the different Part of Speech-Taggers on the Language English for the MSD Tagset

Tagger	Configuration	Time (seconds)		Accuracy	Recall	Precision	F1-Score
		Training	Tagging	(second line: standard deviation)			
nltkBrill	fntbl37	105.7	15.35	0.9263	0.7836	0.8015	0.7924
				0.0023	0.0079	0.0084	0.0080
nltkBrill	brill24	105.5	14.07	0.9214	0.7790	0.7979	0.7883
				0.0026	0.0077	0.0085	0.0080
nltkBrill	nltkdemo18	87.1	14.07	0.9153	0.7760	0.7938	0.7848
				0.0026	0.0080	0.0081	0.0079
nltkBrill	nltkdemo18plus	83.8	14.49	0.9217	0.7809	0.7982	0.7895
				0.0022	0.0083	0.0080	0.0080
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	6.9	0.36	0.9068	0.8142	0.7605	0.7865
				0.0030	0.0084	0.0071	0.0076
skMultinomialNB	[-2, -1, 0, 1, 2]	5.2	0.27	0.9148	0.8273	0.7831	0.8046
				0.0030	0.0077	0.0079	0.0078
skMultinomialNB	[-1, 0, 1]	3.2	0.21	0.9270	0.8415	0.8184	0.8298
				0.0029	0.0075	0.0069	0.0071
skMultinomialNB	[0]	1.3	0.11	0.9177	0.7988	0.8406	0.8192
				0.0018	0.0055	0.0061	0.0057
skMultinomialNB	[-1, 0]	2.3	0.13	0.9115	0.8147	0.8072	0.8109
				0.0020	0.0040	0.0042	0.0038
skMultinomialNB	[-2, -1, 0]	3.3	0.16	0.8990	0.7947	0.7692	0.7817
				0.0022	0.0048	0.0048	0.0046
skMultinomialNB	[-3, -2, -1, 0]	4.2	0.18	0.8921	0.7835	0.7497	0.7662
				0.0031	0.0064	0.0060	0.0061
skMultinomialNB	[0, 1]	2.2	0.16	0.9129	0.7887	0.7789	0.7838
				0.0027	0.0081	0.0082	0.0080
skMultinomialNB	[0, 1, 2]	3.4	0.18	0.8992	0.7687	0.7371	0.7526
				0.0034	0.0089	0.0090	0.0089
skMultinomialNB	[0, 1, 2, 3]	4.2	0.18	0.8918	0.7574	0.7154	0.7358
				0.0034	0.0097	0.0091	0.0094
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	26.1	0.28	0.8708	0.7194	0.6695	0.6935
				0.0052	0.0153	0.0153	0.0153
skPerceptron	[-2, -1, 0, 1, 2]	22.1	0.21	0.8675	0.7154	0.6649	0.6892
				0.0075	0.0173	0.0223	0.0198
skPerceptron	[-1, 0, 1]	16.9	0.11	0.8930	0.7782	0.7493	0.7635
				0.0058	0.0166	0.0171	0.0167
skPerceptron	[0]	11.4	0.05	0.8911	0.7487	0.7879	0.7678
				0.0054	0.0078	0.0084	0.0080
skPerceptron	[-1, 0]	14.5	0.08	0.9051	0.8097	0.7975	0.8035
				0.0042	0.0092	0.0109	0.0098
skPerceptron	[-2, -1, 0]	16.5	0.12	0.8781	0.7490	0.7098	0.7288
				0.0031	0.0112	0.0111	0.0109
skPerceptron	[-3, -2, -1, 0]	19.0	0.18	0.8686	0.7244	0.6782	0.7006
				0.0054	0.0151	0.0153	0.0151
skPerceptron	[0, 1]	12.5	0.06	0.8943	0.8188	0.8143	0.8165
				0.0048	0.0076	0.0085	0.0078
skPerceptron	[0, 1, 2]	15.9	0.11	0.8984	0.8248	0.8095	0.8171
				0.0040	0.0091	0.0100	0.0095
skPerceptron	[0, 1, 2, 3]	17.9	0.16	0.9004	0.8228	0.8064	0.8145
				0.0041	0.0077	0.0085	0.0079
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	22.3	0.23	0.8914	0.7879	0.7640	0.7757
				0.0025	0.0090	0.0093	0.0090
skLinearSVC	[-2, -1, 0, 1, 2]	17.6	0.19	0.9034	0.8165	0.8036	0.8100
				0.0023	0.0087	0.0084	0.0083
skLinearSVC	[-1, 0, 1]	13.9	0.18	0.9079	0.8312	0.8299	0.8305
				0.0021	0.0064	0.0070	0.0066
skLinearSVC	[0]	30.1	0.07	0.9274	0.8313	0.8749	0.8526
				0.0021	0.0057	0.0049	0.0052
skLinearSVC	[-1, 0]	14.7	0.09	0.9316	0.8498	0.8684	0.8590
				0.0018	0.0073	0.0063	0.0067
skLinearSVC	[-2, -1, 0]	16.2	0.10	0.9255	0.8374	0.8460	0.8417
				0.0025	0.0093	0.0088	0.0090
skLinearSVC	[-3, -2, -1, 0]	21.6	0.18	0.9187	0.8214	0.8221	0.8218
				0.0031	0.0106	0.0105	0.0105
skLinearSVC	[0, 1]	12.3	0.09	0.9023	0.8387	0.8544	0.8465
				0.0021	0.0060	0.0050	0.0054
skLinearSVC	[0, 1, 2]	14.0	0.11	0.9035	0.8395	0.8507	0.8451
				0.0024	0.0060	0.0057	0.0057
skLinearSVC	[0, 1, 2, 3]	15.3	0.11	0.9006	0.8344	0.8416	0.8380
				0.0028	0.0067	0.0055	0.0060

Table 7.2: Results of the different Part of Speech-Taggers on the Language English for the Universal Tagset

Tagger	**Configuration**	**Time (seconds)**		**Accuracy**	**Recall**	**Precision**	**F1-Score**
		Training	**Tagging**	(second line: standard deviation)			
nltkBrill	fntbl37	155.3	13.83	0.8611	0.7224	0.7667	0.7439
				0.0062	0.0109	0.0111	0.0109
nltkBrill	brill24	136.8	12.50	0.8586	0.7228	0.7650	0.7433
				0.0063	0.0107	0.0112	0.0109
nltkBrill	nltkdemo18	102.3	11.93	0.8576	0.7220	0.7631	0.7420
				0.0064	0.0104	0.0110	0.0107
nltkBrill	nltkdemo18plus	113.2	11.82	0.8577	0.7218	0.7633	0.7420
				0.0062	0.0103	0.0110	0.0106
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	13.1	0.71	0.6791	0.5526	0.4748	0.5107
				0.0055	0.0074	0.0064	0.0067
skMultinomialNB	[-2, -1, 0, 1, 2]	9.7	0.54	0.7132	0.5779	0.5045	0.5387
				0.0052	0.0063	0.0059	0.0059
skMultinomialNB	[-1, 0, 1]	6.4	0.35	0.7490	0.6047	0.5509	0.5766
				0.0059	0.0070	0.0071	0.0066
skMultinomialNB	[0]	3.2	0.24	0.7433	0.5651	0.6207	0.5916
				0.0064	0.0036	0.0056	0.0044
skMultinomialNB	[-1, 0]	4.7	0.35	0.7446	0.5777	0.5421	0.5593
				0.0060	0.0058	0.0075	0.0065
skMultinomialNB	[-2, -1, 0]	6.1	0.43	0.7164	0.5525	0.4938	0.5215
				0.0056	0.0070	0.0068	0.0068
skMultinomialNB	[-3, -2, -1, 0]	7.9	0.45	0.6938	0.5340	0.4665	0.4980
				0.0054	0.0064	0.0059	0.0060
skMultinomialNB	[0, 1]	4.6	0.34	0.7309	0.5811	0.5612	0.5710
				0.0058	0.0051	0.0065	0.0056
skMultinomialNB	[0, 1, 2]	6.3	0.40	0.7095	0.5636	0.5206	0.5412
				0.0056	0.0048	0.0064	0.0056
skMultinomialNB	[0, 1, 2, 3]	7.8	0.47	0.6885	0.5467	0.4938	0.5189
				0.0060	0.0053	0.0073	0.0063
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	728.2	0.61	0.7929	0.6266	0.6159	0.6212
				0.0046	0.0110	0.0113	0.0111
skPerceptron	[-2, -1, 0, 1, 2]	641.8	0.39	0.7987	0.6450	0.6344	0.6396
				0.0053	0.0108	0.0098	0.0101
skPerceptron	[-1, 0, 1]	536.2	0.29	0.7973	0.6599	0.6502	0.6550
				0.0053	0.0110	0.0102	0.0104
skPerceptron	[0]	324.0	0.15	0.7881	0.6775	0.7442	0.7093
				0.0117	0.0106	0.0115	0.0109
skPerceptron	[-1, 0]	452.5	0.20	0.7918	0.6766	0.6628	0.6696
				0.0069	0.0096	0.0105	0.0098
skPerceptron	[-2, -1, 0]	526.6	0.30	0.7979	0.6756	0.6576	0.6665
				0.0049	0.0116	0.0105	0.0109
skPerceptron	[-3, -2, -1, 0]	573.5	0.35	0.7984	0.6758	0.6531	0.6643
				0.0051	0.0106	0.0086	0.0095
skPerceptron	[0, 1]	441.4	0.24	0.7790	0.6915	0.6804	0.6859
				0.0132	0.0115	0.0125	0.0119
skPerceptron	[0, 1, 2]	512.5	0.29	0.7712	0.6719	0.6427	0.6569
				0.0085	0.0098	0.0119	0.0105
skPerceptron	[0, 1, 2, 3]	559.0	0.35	0.7607	0.6522	0.6093	0.6300
				0.0057	0.0070	0.0063	0.0063
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	197.3	0.39	0.8243	0.6980	0.7098	0.7039
				0.0043	0.0067	0.0064	0.0062
skLinearSVC	[-2, -1, 0, 1, 2]	205.4	0.26	0.8366	0.7274	0.7438	0.7355
				0.0039	0.0067	0.0070	0.0066
skLinearSVC	[-1, 0, 1]	255.2	0.17	0.8489	0.7547	0.7793	0.7668
				0.0039	0.0080	0.0074	0.0076
skLinearSVC	[0]	938.8	0.10	0.8366	0.7292	0.8009	0.7634
				0.0052	0.0079	0.0087	0.0081
skLinearSVC	[-1, 0]	469.5	0.12	0.8483	0.7532	0.7895	0.7709
				0.0054	0.0117	0.0118	0.0117
skLinearSVC	[-2, -1, 0]	277.7	0.16	0.8482	0.7499	0.7805	0.7649
				0.0046	0.0089	0.0094	0.0089
skLinearSVC	[-3, -2, -1, 0]	242.3	0.27	0.8443	0.7403	0.7690	0.7543
				0.0050	0.0080	0.0089	0.0082
skLinearSVC	[0, 1]	518.4	0.12	0.8347	0.7475	0.7916	0.7689
				0.0050	0.0076	0.0085	0.0078
skLinearSVC	[0, 1, 2]	334.9	0.15	0.8307	0.7478	0.7860	0.7664
				0.0049	0.0073	0.0086	0.0077
skLinearSVC	[0, 1, 2, 3]	290.3	0.22	0.8255	0.7353	0.7696	0.7521
				0.0052	0.0088	0.0098	0.0092

Table 7.3: Results of the different Part of Speech-Taggers on the Language Farsi for the MSD Tagset

Tagger	Configuration	Time (seconds)		Accuracy	Recall	Precision	F1-Score
		Training	Tagging	(second line: standard deviation)			
nltkBrill	fntbl37	75.3	11.25	0.9131	0.7793	0.7935	0.7863
				0.0049	0.0117	0.0121	0.0118
nltkBrill	brill24	74.2	11.22	0.9109	0.7770	0.7915	0.7841
				0.0050	0.0116	0.0120	0.0118
nltkBrill	nltkdemo18	61.0	10.78	0.9083	0.7745	0.7892	0.7818
				0.0051	0.0117	0.0121	0.0119
nltkBrill	nltkdemo18plus	62.3	10.92	0.9108	0.7764	0.7910	0.7836
				0.0046	0.0120	0.0121	0.0120
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	6.3	0.35	0.9010	0.8226	0.7796	0.8005
				0.0033	0.0067	0.0068	0.0066
skMultinomialNB	[-2, -1, 0, 1, 2]	4.9	0.22	0.9131	0.8362	0.8047	0.8201
				0.0027	0.0078	0.0059	0.0067
skMultinomialNB	[-1, 0, 1]	3.2	0.13	0.9254	0.8472	0.8343	0.8407
				0.0029	0.0046	0.0056	0.0050
skMultinomialNB	[0]	1.3	0.08	0.9327	0.8621	0.8929	0.8772
				0.0019	0.0043	0.0051	0.0046
skMultinomialNB	[-1, 0]	2.3	0.14	0.9233	0.8506	0.8451	0.8478
				0.0033	0.0050	0.0060	0.0054
skMultinomialNB	[-2, -1, 0]	3.1	0.18	0.9108	0.8369	0.8153	0.8259
				0.0042	0.0069	0.0062	0.0064
skMultinomialNB	[-3, -2, -1, 0]	3.9	0.19	0.9010	0.8246	0.7931	0.8085
				0.0039	0.0057	0.0055	0.0054
skMultinomialNB	[0, 1]	2.2	0.15	0.9105	0.8080	0.8050	0.8065
				0.0027	0.0059	0.0058	0.0057
skMultinomialNB	[0, 1, 2]	3.1	0.19	0.8971	0.7844	0.7689	0.7766
				0.0028	0.0056	0.0051	0.0052
skMultinomialNB	[0, 1, 2, 3]	3.9	0.20	0.8886	0.7721	0.7485	0.7601
				0.0033	0.0063	0.0061	0.0061
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	21.1	0.17	0.8633	0.7267	0.6808	0.7030
				0.0033	0.0080	0.0080	0.0079
skPerceptron	[-2, -1, 0, 1, 2]	17.8	0.16	0.8777	0.7525	0.7163	0.7339
				0.0038	0.0084	0.0098	0.0088
skPerceptron	[-1, 0, 1]	13.8	0.11	0.8912	0.7879	0.7591	0.7732
				0.0051	0.0102	0.0114	0.0106
skPerceptron	[0]	7.7	0.04	0.9203	0.8657	0.8965	0.8808
				0.0106	0.0041	0.0041	0.0039
skPerceptron	[-1, 0]	11.4	0.08	0.9245	0.8635	0.8663	0.8649
				0.0035	0.0050	0.0045	0.0047
skPerceptron	[-2, -1, 0]	13.6	0.08	0.9194	0.8541	0.8490	0.8515
				0.0019	0.0045	0.0055	0.0048
skPerceptron	[-3, -2, -1, 0]	15.6	0.10	0.9115	0.8431	0.8335	0.8383
				0.0047	0.0054	0.0055	0.0053
skPerceptron	[0, 1]	11.2	0.05	0.9221	0.8563	0.8569	0.8566
				0.0029	0.0059	0.0068	0.0063
skPerceptron	[0, 1, 2]	14.0	0.10	0.9142	0.8387	0.8305	0.8346
				0.0040	0.0058	0.0060	0.0058
skPerceptron	[0, 1, 2, 3]	16.1	0.14	0.9022	0.8107	0.7961	0.8033
				0.0038	0.0091	0.0087	0.0087
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	20.3	0.23	0.8712	0.7457	0.7125	0.7287
				0.0030	0.0076	0.0064	0.0068
skLinearSVC	[-2, -1, 0, 1, 2]	17.8	0.19	0.8783	0.7555	0.7286	0.7418
				0.0033	0.0066	0.0050	0.0054
skLinearSVC	[-1, 0, 1]	10.7	0.11	0.8921	0.7782	0.7616	0.7698
				0.0036	0.0078	0.0071	0.0073
skLinearSVC	[0]	27.9	0.06	0.9376	0.8721	0.9032	0.8874
				0.0015	0.0039	0.0042	0.0039
skLinearSVC	[-1, 0]	12.4	0.08	0.9357	0.8747	0.8893	0.8819
				0.0017	0.0055	0.0043	0.0048
skLinearSVC	[-2, -1, 0]	13.1	0.10	0.9317	0.8676	0.8776	0.8726
				0.0016	0.0042	0.0037	0.0038
skLinearSVC	[-3, -2, -1, 0]	17.3	0.19	0.9281	0.8597	0.8678	0.8637
				0.0018	0.0044	0.0045	0.0043
skLinearSVC	[0, 1]	13.9	0.12	0.8817	0.7517	0.7519	0.7518
				0.0055	0.0109	0.0104	0.0106
skLinearSVC	[0, 1, 2]	14.4	0.14	0.8659	0.7217	0.7110	0.7163
				0.0051	0.0094	0.0092	0.0093
skLinearSVC	[0, 1, 2, 3]	15.8	0.13	0.8516	0.6974	0.6790	0.6881
				0.0049	0.0100	0.0097	0.0098

Table 7.4: Results of the different Part of Speech-Taggers on the Language Farsi for the Universal Tagset

Tagger	Configuration	Time (seconds) Training	Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	60.7	12.88	0.8837	0.7262	0.7383	0.7322
				0.0035	0.0086	0.0083	0.0084
nltkBrill	brill24	52.4	12.69	0.8841	0.7270	0.7389	0.7329
				0.0034	0.0080	0.0080	0.0080
nltkBrill	nltkdemo18	43.7	12.94	0.8836	0.7272	0.7387	0.7329
				0.0035	0.0083	0.0082	0.0082
nltkBrill	nltkdemo18plus	48.8	12.60	0.8836	0.7272	0.7387	0.7329
				0.0035	0.0083	0.0082	0.0082
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	13.0	0.75	0.6410	0.4757	0.3864	0.4264
				0.0047	0.0069	0.0069	0.0069
skMultinomialNB	[-2, -1, 0, 1, 2]	10.0	0.62	0.6848	0.5103	0.4243	0.4633
				0.0053	0.0090	0.0082	0.0085
skMultinomialNB	[-1, 0, 1]	6.8	0.46	0.7294	0.5324	0.4641	0.4959
				0.0048	0.0094	0.0087	0.0090
skMultinomialNB	[0]	3.2	0.25	0.7011	0.4136	0.4304	0.4219
				0.0032	0.0073	0.0071	0.0072
skMultinomialNB	[-1, 0]	5.0	0.34	0.7329	0.5320	0.4829	0.5063
				0.0026	0.0078	0.0075	0.0075
skMultinomialNB	[-2, -1, 0]	6.6	0.42	0.7051	0.5190	0.4514	0.4828
				0.0038	0.0067	0.0065	0.0066
skMultinomialNB	[-3, -2, -1, 0]	8.4	0.58	0.6758	0.4969	0.4215	0.4561
				0.0031	0.0059	0.0055	0.0057
skMultinomialNB	[0, 1]	4.9	0.33	0.7013	0.4447	0.4017	0.4221
				0.0046	0.0079	0.0075	0.0076
skMultinomialNB	[0, 1, 2]	6.6	0.44	0.6622	0.4161	0.3578	0.3847
				0.0039	0.0071	0.0072	0.0071
skMultinomialNB	[0, 1, 2, 3]	8.3	0.51	0.6306	0.3932	0.3292	0.3583
				0.0038	0.0075	0.0072	0.0074
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	707.8	0.56	0.8261	0.6392	0.6235	0.6313
				0.0032	0.0067	0.0069	0.0067
skPerceptron	[-2, -1, 0, 1, 2]	628.3	0.47	0.8375	0.6638	0.6523	0.6580
				0.0039	0.0076	0.0084	0.0079
skPerceptron	[-1, 0, 1]	513.3	0.31	0.8330	0.6779	0.6707	0.6743
				0.0070	0.0099	0.0106	0.0102
skPerceptron	[0]	315.2	0.16	0.8421	0.6915	0.7196	0.7052
				0.0062	0.0091	0.0094	0.0092
skPerceptron	[-1, 0]	425.2	0.23	0.8219	0.6662	0.6522	0.6592
				0.0046	0.0129	0.0139	0.0132
skPerceptron	[-2, -1, 0]	501.2	0.32	0.8382	0.6772	0.6679	0.6725
				0.0037	0.0114	0.0131	0.0122
skPerceptron	[-3, -2, -1, 0]	556.4	0.46	0.8465	0.6851	0.6775	0.6813
				0.0043	0.0093	0.0102	0.0097
skPerceptron	[0, 1]	460.0	0.25	0.8266	0.6877	0.6865	0.6871
				0.0058	0.0087	0.0087	0.0086
skPerceptron	[0, 1, 2]	528.6	0.31	0.8252	0.6745	0.6675	0.6710
				0.0045	0.0091	0.0098	0.0094
skPerceptron	[0, 1, 2, 3]	635.8	0.44	0.8175	0.6599	0.6480	0.6539
				0.0045	0.0065	0.0057	0.0060
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	222.1	0.41	0.8446	0.7089	0.7065	0.7077
				0.0037	0.0086	0.0085	0.0085
skLinearSVC	[-2, -1, 0, 1, 2]	218.6	0.30	0.8580	0.7400	0.7426	0.7413
				0.0036	0.0099	0.0099	0.0099
skLinearSVC	[-1, 0, 1]	294.6	0.18	0.8685	0.7621	0.7712	0.7666
				0.0037	0.0079	0.0080	0.0079
skLinearSVC	[0]	839.8	0.12	0.8639	0.7154	0.7444	0.7296
				0.0035	0.0085	0.0083	0.0084
skLinearSVC	[-1, 0]	673.5	0.14	0.8778	0.7459	0.7574	0.7516
				0.0032	0.0080	0.0072	0.0076
skLinearSVC	[-2, -1, 0]	393.5	0.19	0.8799	0.7513	0.7601	0.7557
				0.0030	0.0089	0.0096	0.0092
skLinearSVC	[-3, -2, -1, 0]	322.6	0.23	0.8756	0.7424	0.7490	0.7457
				0.0026	0.0075	0.0077	0.0076
skLinearSVC	[0, 1]	628.4	0.13	0.8531	0.7327	0.7492	0.7409
				0.0047	0.0082	0.0076	0.0078
skLinearSVC	[0, 1, 2]	374.5	0.18	0.8510	0.7323	0.7460	0.7391
				0.0040	0.0079	0.0070	0.0074
skLinearSVC	[0, 1, 2, 3]	302.6	0.26	0.8490	0.7284	0.7415	0.7349
				0.0044	0.0090	0.0080	0.0084

Table 7.5: Results of the different Part of Speech-Taggers on the Language Romanian for the MSD Tagset

Tagger	Configuration	Time (seconds)		Accuracy	Recall	Precision	F1-Score
		Training	Tagging	(second line: standard deviation)			
nltkBrill	fntbl37	67.4	12.69	0.8947	0.7400	0.7475	0.7437
				0.0030	0.0070	0.0071	0.0070
nltkBrill	brill24	71.8	12.29	0.8912	0.7374	0.7452	0.7413
				0.0031	0.0074	0.0068	0.0071
nltkBrill	nltkdemo18	57.3	12.61	0.8883	0.7355	0.7432	0.7394
				0.0026	0.0072	0.0063	0.0067
nltkBrill	nltkdemo18plus	59.4	12.64	0.8907	0.7370	0.7450	0.7410
				0.0026	0.0069	0.0066	0.0067
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	7.2	0.35	0.8886	0.8087	0.7623	0.7848
				0.0033	0.0069	0.0072	0.0069
skMultinomialNB	[-2, -1, 0, 1, 2]	5.0	0.32	0.9028	0.8236	0.7885	0.8057
				0.0029	0.0068	0.0071	0.0069
skMultinomialNB	[-1, 0, 1]	3.4	0.18	0.9151	0.8361	0.8164	0.8261
				0.0028	0.0062	0.0069	0.0065
skMultinomialNB	[0]	1.4	0.09	0.9037	0.8116	0.8338	0.8226
				0.0028	0.0047	0.0042	0.0044
skMultinomialNB	[-1, 0]	2.4	0.14	0.9097	0.8318	0.8233	0.8275
				0.0035	0.0076	0.0086	0.0081
skMultinomialNB	[-2, -1, 0]	3.5	0.16	0.8971	0.8139	0.7929	0.8032
				0.0032	0.0060	0.0062	0.0060
skMultinomialNB	[-3, -2, -1, 0]	4.4	0.27	0.8857	0.8002	0.7705	0.7851
				0.0033	0.0069	0.0069	0.0068
skMultinomialNB	[0, 1]	2.5	0.14	0.8944	0.7857	0.7770	0.7813
				0.0021	0.0034	0.0036	0.0034
skMultinomialNB	[0, 1, 2]	3.4	0.17	0.8818	0.7656	0.7447	0.7550
				0.0032	0.0067	0.0080	0.0073
skMultinomialNB	[0, 1, 2, 3]	4.0	0.20	0.8710	0.7507	0.7220	0.7360
				0.0040	0.0090	0.0103	0.0096
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	27.2	0.22	0.8782	0.7632	0.7345	0.7486
				0.0050	0.0083	0.0108	0.0095
skPerceptron	[-2, -1, 0, 1, 2]	21.7	0.17	0.8823	0.7715	0.7460	0.7585
				0.0054	0.0114	0.0118	0.0115
skPerceptron	[-1, 0, 1]	17.0	0.07	0.8706	0.7620	0.7356	0.7486
				0.0068	0.0120	0.0137	0.0128
skPerceptron	[0]	9.6	0.05	0.8649	0.7191	0.7388	0.7288
				0.0063	0.0078	0.0080	0.0079
skPerceptron	[-1, 0]	14.2	0.07	0.8292	0.6775	0.6478	0.6623
				0.0101	0.0152	0.0164	0.0157
skPerceptron	[-2, -1, 0]	17.3	0.10	0.8424	0.6980	0.6654	0.6813
				0.0086	0.0154	0.0179	0.0167
skPerceptron	[-3, -2, -1, 0]	20.6	0.13	0.8608	0.7319	0.7028	0.7170
				0.0059	0.0103	0.0115	0.0109
skPerceptron	[0, 1]	14.4	0.05	0.8958	0.8110	0.8081	0.8095
				0.0044	0.0080	0.0087	0.0083
skPerceptron	[0, 1, 2]	16.9	0.09	0.8876	0.7975	0.7865	0.7920
				0.0034	0.0057	0.0071	0.0063
skPerceptron	[0, 1, 2, 3]	19.0	0.12	0.8830	0.7821	0.7670	0.7745
				0.0040	0.0068	0.0061	0.0062
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	27.0	0.32	0.9117	0.8281	0.8151	0.8215
				0.0048	0.0071	0.0090	0.0080
skLinearSVC	[-2, -1, 0, 1, 2]	20.9	0.20	0.9169	0.8375	0.8274	0.8324
				0.0045	0.0073	0.0079	0.0075
skLinearSVC	[-1, 0, 1]	15.9	0.21	0.9192	0.8393	0.8358	0.8376
				0.0041	0.0069	0.0074	0.0071
skLinearSVC	[0]	32.3	0.04	0.9123	0.8222	0.8446	0.8332
				0.0035	0.0056	0.0051	0.0053
skLinearSVC	[-1, 0]	16.8	0.06	0.9089	0.8207	0.8184	0.8195
				0.0033	0.0068	0.0076	0.0071
skLinearSVC	[-2, -1, 0]	19.8	0.18	0.9124	0.8263	0.8222	0.8242
				0.0036	0.0059	0.0067	0.0062
skLinearSVC	[-3, -2, -1, 0]	23.8	0.17	0.9085	0.8195	0.8131	0.8163
				0.0038	0.0063	0.0074	0.0067
skLinearSVC	[0, 1]	17.9	0.15	0.9184	0.8384	0.8497	0.8440
				0.0032	0.0046	0.0050	0.0047
skLinearSVC	[0, 1, 2]	19.9	0.14	0.9179	0.8410	0.8494	0.8452
				0.0029	0.0056	0.0049	0.0052
skLinearSVC	[0, 1, 2, 3]	23.7	0.21	0.9155	0.8358	0.8425	0.8392
				0.0033	0.0065	0.0057	0.0060

Table 7.6: Results of the different Part of Speech-Taggers on the Language Romanian for the Universal Tagset

Tagger	Configuration	Time (seconds) Training	Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	93.2	12.25	0.7901 0.0049	0.5854 0.0073	0.6146 0.0071	0.5996 0.0072
nltkBrill	brill24	80.8	12.03	0.7901 0.0049	0.5859 0.0076	0.6148 0.0078	0.6000 0.0076
nltkBrill	nltkdemo18	64.1	11.53	0.7899 0.0050	0.5859 0.0075	0.6148 0.0077	0.6000 0.0076
nltkBrill	nltkdemo18plus	70.8	11.84	0.7899 0.0050	0.5859 0.0075	0.6148 0.0077	0.6000 0.0076
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	21.4	1.82	0.5029 0.0045	0.2160 0.0045	0.1936 0.0044	0.2042 0.0044
skMultinomialNB	[-2, -1, 0, 1, 2]	17.1	1.26	0.5349 0.0048	0.2430 0.0050	0.2187 0.0044	0.2302 0.0047
skMultinomialNB	[-1, 0, 1]	11.0	0.91	0.5732 0.0050	0.2737 0.0059	0.2521 0.0051	0.2625 0.0054
skMultinomialNB	[0]	4.9	0.50	0.5783 0.0042	0.2383 0.0047	0.2577 0.0048	0.2476 0.0048
skMultinomialNB	[-1, 0]	8.0	0.68	0.5886 0.0053	0.2891 0.0076	0.2789 0.0073	0.2839 0.0074
skMultinomialNB	[-2, -1, 0]	11.1	0.91	0.5664 0.0042	0.2726 0.0063	0.2552 0.0060	0.2636 0.0061
skMultinomialNB	[-3, -2, -1, 0]	13.3	1.08	0.5432 0.0041	0.2509 0.0057	0.2319 0.0051	0.2410 0.0054
skMultinomialNB	[0, 1]	8.1	0.74	0.5591 0.0051	0.2385 0.0059	0.2255 0.0053	0.2318 0.0056
skMultinomialNB	[0, 1, 2]	11.0	0.85	0.5341 0.0047	0.2217 0.0063	0.2013 0.0055	0.2110 0.0059
skMultinomialNB	[0, 1, 2, 3]	13.2	1.18	0.5158 0.0043	0.2085 0.0052	0.1864 0.0043	0.1968 0.0047
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	1560.5	1.83	0.7195 0.0051	0.4778 0.0066	0.4779 0.0077	0.4779 0.0071
skPerceptron	[-2, -1, 0, 1, 2]	1367.8	1.22	0.7338 0.0063	0.4995 0.0103	0.5037 0.0104	0.5016 0.0104
skPerceptron	[-1, 0, 1]	1096.5	0.79	0.7383 0.0054	0.5181 0.0087	0.5241 0.0089	0.5211 0.0087
skPerceptron	[0]	680.6	0.42	0.7384 0.0097	0.5357 0.0075	0.5794 0.0081	0.5567 0.0078
skPerceptron	[-1, 0]	901.5	0.61	0.7309 0.0046	0.5225 0.0096	0.5306 0.0107	0.5265 0.0100
skPerceptron	[-2, -1, 0]	1048.3	0.81	0.7359 0.0048	0.5193 0.0072	0.5267 0.0068	0.5230 0.0069
skPerceptron	[-3, -2, -1, 0]	1171.3	0.91	0.7346 0.0057	0.5154 0.0100	0.5230 0.0104	0.5192 0.0101
skPerceptron	[0, 1]	878.1	0.62	0.7302 0.0057	0.5247 0.0067	0.5300 0.0078	0.5274 0.0071
skPerceptron	[0, 1, 2]	1051.0	0.85	0.7272 0.0052	0.5134 0.0068	0.5123 0.0074	0.5128 0.0070
skPerceptron	[0, 1, 2, 3]	1183.4	1.00	0.7157 0.0057	0.4936 0.0093	0.4855 0.0102	0.4895 0.0097
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	401.1	0.62	0.7498 0.0055	0.5232 0.0101	0.5337 0.0103	0.5284 0.0101
skLinearSVC	[-2, -1, 0, 1, 2]	405.5	0.44	0.7687 0.0056	0.5607 0.0093	0.5755 0.0100	0.5680 0.0096
skLinearSVC	[-1, 0, 1]	534.1	0.34	0.7836 0.0051	0.5917 0.0092	0.6126 0.0089	0.6020 0.0090
skLinearSVC	[0]	1835.9	0.23	0.7694 0.0047	0.5621 0.0078	0.6080 0.0083	0.5842 0.0080
skLinearSVC	[-1, 0]	1228.6	0.25	0.7801 0.0049	0.5897 0.0075	0.6163 0.0074	0.6027 0.0074
skLinearSVC	[-2, -1, 0]	655.5	0.35	0.7757 0.0060	0.5801 0.0094	0.6047 0.0097	0.5921 0.0095
skLinearSVC	[-3, -2, -1, 0]	561.6	0.34	0.7675 0.0050	0.5653 0.0081	0.5876 0.0080	0.5762 0.0080
skLinearSVC	[0, 1]	1101.1	0.28	0.7761 0.0050	0.5801 0.0078	0.6084 0.0084	0.5939 0.0081
skLinearSVC	[0, 1, 2]	639.1	0.32	0.7762 0.0047	0.5797 0.0076	0.6052 0.0081	0.5922 0.0078
skLinearSVC	[0, 1, 2, 3]	537.6	0.35	0.7718 0.0047	0.5714 0.0078	0.5960 0.0084	0.5834 0.0081

Table 7.7: Results of the different Part of Speech-Taggers on the Language Slovenian for the MSD Tagset

Tagger	Configuration	Time (seconds) Training	Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	35.1	10.49	0.8579	0.6746	0.6802	0.6774
				0.0046	0.0081	0.0082	0.0081
nltkBrill	brill24	28.2	10.14	0.8572	0.6730	0.6796	0.6763
				0.0047	0.0080	0.0080	0.0080
nltkBrill	nltkdemo18	17.8	10.25	0.8561	0.6716	0.6788	0.6752
				0.0047	0.0079	0.0080	0.0079
nltkBrill	nltkdemo18plus	25.4	10.79	0.8568	0.6726	0.6791	0.6758
				0.0046	0.0079	0.0081	0.0080
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	6.3	0.31	0.8451	0.7342	0.6859	0.7093
				0.0039	0.0066	0.0078	0.0071
skMultinomialNB	[-2, -1, 0, 1, 2]	4.9	0.24	0.8591	0.7473	0.7073	0.7267
				0.0034	0.0062	0.0074	0.0067
skMultinomialNB	[-1, 0, 1]	3.1	0.17	0.8761	0.7593	0.7355	0.7472
				0.0029	0.0052	0.0061	0.0056
skMultinomialNB	[0]	1.2	0.12	0.8849	0.7427	0.7529	0.7477
				0.0042	0.0073	0.0073	0.0073
skMultinomialNB	[-1, 0]	2.2	0.14	0.8759	0.7469	0.7348	0.7408
				0.0037	0.0066	0.0069	0.0067
skMultinomialNB	[-2, -1, 0]	3.1	0.13	0.8651	0.7407	0.7159	0.7281
				0.0044	0.0076	0.0082	0.0079
skMultinomialNB	[-3, -2, -1, 0]	3.9	0.17	0.8556	0.7316	0.7003	0.7156
				0.0044	0.0068	0.0084	0.0075
skMultinomialNB	[0, 1]	2.2	0.12	0.8714	0.7339	0.7209	0.7273
				0.0039	0.0058	0.0071	0.0064
skMultinomialNB	[0, 1, 2]	2.9	0.21	0.8587	0.7177	0.6942	0.7058
				0.0030	0.0052	0.0058	0.0054
skMultinomialNB	[0, 1, 2, 3]	3.9	0.19	0.8484	0.7056	0.6756	0.6903
				0.0035	0.0057	0.0063	0.0059
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	21.4	0.21	0.8583	0.7093	0.6896	0.6993
				0.0037	0.0066	0.0072	0.0068
skPerceptron	[-2, -1, 0, 1, 2]	18.6	0.14	0.8761	0.7408	0.7271	0.7339
				0.0042	0.0086	0.0092	0.0088
skPerceptron	[-1, 0, 1]	15.0	0.14	0.8856	0.7593	0.7496	0.7544
				0.0051	0.0113	0.0115	0.0114
skPerceptron	[0]	9.0	0.05	0.8514	0.6701	0.6793	0.6747
				0.0047	0.0070	0.0071	0.0071
skPerceptron	[-1, 0]	12.2	0.10	0.8618	0.7062	0.6987	0.7024
				0.0050	0.0086	0.0085	0.0085
skPerceptron	[-2, -1, 0]	14.2	0.08	0.8582	0.7004	0.6892	0.6948
				0.0058	0.0112	0.0109	0.0110
skPerceptron	[-3, -2, -1, 0]	16.9	0.12	0.8616	0.7101	0.6977	0.7038
				0.0051	0.0116	0.0116	0.0116
skPerceptron	[0, 1]	13.5	0.07	0.8911	0.7698	0.7677	0.7687
				0.0084	0.0170	0.0167	0.0168
skPerceptron	[0, 1, 2]	15.3	0.10	0.8881	0.7654	0.7588	0.7621
				0.0050	0.0102	0.0103	0.0102
skPerceptron	[0, 1, 2, 3]	16.6	0.13	0.8834	0.7581	0.7482	0.7531
				0.0031	0.0062	0.0059	0.0060
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	29.8	0.23	0.8778	0.7465	0.7342	0.7403
				0.0039	0.0073	0.0071	0.0071
skLinearSVC	[-2, -1, 0, 1, 2]	23.6	0.17	0.9001	0.7880	0.7827	0.7854
				0.0046	0.0102	0.0098	0.0100
skLinearSVC	[-1, 0, 1]	16.3	0.13	0.9170	0.8213	0.8217	0.8215
				0.0041	0.0086	0.0088	0.0087
skLinearSVC	[0]	25.0	0.06	0.9024	0.7812	0.7919	0.7865
				0.0037	0.0070	0.0071	0.0070
skLinearSVC	[-1, 0]	18.3	0.06	0.9055	0.7922	0.7967	0.7945
				0.0033	0.0070	0.0069	0.0069
skLinearSVC	[-2, -1, 0]	21.0	0.13	0.9004	0.7827	0.7846	0.7837
				0.0031	0.0063	0.0065	0.0064
skLinearSVC	[-3, -2, -1, 0]	24.0	0.14	0.8934	0.7697	0.7688	0.7692
				0.0035	0.0071	0.0071	0.0071
skLinearSVC	[0, 1]	16.6	0.11	0.9070	0.7948	0.8011	0.7979
				0.0032	0.0061	0.0067	0.0064
skLinearSVC	[0, 1, 2]	20.0	0.15	0.9049	0.7912	0.7957	0.7934
				0.0041	0.0074	0.0083	0.0078
skLinearSVC	[0, 1, 2, 3]	22.0	0.11	0.9007	0.7832	0.7860	0.7846
				0.0041	0.0080	0.0085	0.0082

Table 7.8: Results of the different Part of Speech-Taggers on the Language Slovenian for the Universal Tagset

Tagger	Configuration	Time (seconds) Training	Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	98.8	11.21	0.7318	0.5583	0.5881	0.5728
				0.0057	0.0064	0.0064	0.0064
nltkBrill	brill24	79.5	10.53	0.7324	0.5592	0.5890	0.5737
				0.0057	0.0062	0.0062	0.0062
nltkBrill	nltkdemo18	64.8	10.65	0.7322	0.5592	0.5888	0.5736
				0.0056	0.0063	0.0062	0.0062
nltkBrill	nltkdemo18plus	72.6	10.33	0.7322	0.5592	0.5888	0.5736
				0.0056	0.0063	0.0062	0.0062
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	22.3	2.01	0.4431	0.2537	0.2334	0.2431
				0.0047	0.0050	0.0044	0.0047
skMultinomialNB	[-2, -1, 0, 1, 2]	17.7	1.41	0.4775	0.2800	0.2581	0.2686
				0.0062	0.0071	0.0064	0.0067
skMultinomialNB	[-1, 0, 1]	11.6	0.93	0.5166	0.3074	0.2897	0.2983
				0.0060	0.0071	0.0067	0.0069
skMultinomialNB	[0]	5.2	0.51	0.4962	0.2378	0.2578	0.2474
				0.0062	0.0059	0.0061	0.0060
skMultinomialNB	[-1, 0]	8.5	0.82	0.5202	0.3019	0.2935	0.2976
				0.0056	0.0064	0.0060	0.0062
skMultinomialNB	[-2, -1, 0]	11.2	1.04	0.5012	0.2930	0.2779	0.2853
				0.0054	0.0051	0.0045	0.0048
skMultinomialNB	[-3, -2, -1, 0]	14.3	1.21	0.4798	0.2786	0.2616	0.2698
				0.0049	0.0054	0.0047	0.0050
skMultinomialNB	[0, 1]	8.4	0.77	0.4936	0.2607	0.2550	0.2578
				0.0067	0.0058	0.0055	0.0057
skMultinomialNB	[0, 1, 2]	10.6	0.98	0.4674	0.2445	0.2314	0.2378
				0.0057	0.0053	0.0050	0.0052
skMultinomialNB	[0, 1, 2, 3]	13.6	1.15	0.4478	0.2318	0.2176	0.2245
				0.0049	0.0055	0.0052	0.0053
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	1430.8	1.75	0.6429	0.4368	0.4352	0.4360
				0.0074	0.0084	0.0097	0.0090
skPerceptron	[-2, -1, 0, 1, 2]	1170.3	1.19	0.6574	0.4600	0.4626	0.4613
				0.0072	0.0082	0.0094	0.0088
skPerceptron	[-1, 0, 1]	886.0	0.79	0.6675	0.4851	0.4925	0.4888
				0.0064	0.0106	0.0103	0.0104
skPerceptron	[0]	529.6	0.33	0.6782	0.5076	0.5505	0.5282
				0.0078	0.0077	0.0077	0.0077
skPerceptron	[-1, 0]	753.8	0.57	0.6591	0.4895	0.4982	0.4938
				0.0089	0.0106	0.0111	0.0108
skPerceptron	[-2, -1, 0]	863.1	0.80	0.6677	0.4891	0.4976	0.4933
				0.0079	0.0117	0.0121	0.0119
skPerceptron	[-3, -2, -1, 0]	967.7	0.84	0.6678	0.4844	0.4919	0.4881
				0.0071	0.0096	0.0100	0.0098
skPerceptron	[0, 1]	708.8	0.62	0.6644	0.4983	0.5044	0.5013
				0.0082	0.0091	0.0105	0.0097
skPerceptron	[0, 1, 2]	857.9	0.79	0.6570	0.4871	0.4863	0.4867
				0.0071	0.0084	0.0088	0.0085
skPerceptron	[0, 1, 2, 3]	957.9	0.90	0.6487	0.4706	0.4642	0.4674
				0.0055	0.0072	0.0079	0.0075
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	335.5	0.41	0.6801	0.4904	0.4980	0.4942
				0.0072	0.0094	0.0103	0.0098
skLinearSVC	[-2, -1, 0, 1, 2]	355.5	0.31	0.6988	0.5201	0.5325	0.5262
				0.0078	0.0098	0.0104	0.0101
skLinearSVC	[-1, 0, 1]	456.4	0.29	0.7191	0.5525	0.5723	0.5622
				0.0070	0.0094	0.0093	0.0093
skLinearSVC	[0]	1542.1	0.18	0.7134	0.5421	0.5879	0.5641
				0.0055	0.0074	0.0072	0.0072
skLinearSVC	[-1, 0]	1067.8	0.22	0.7177	0.5570	0.5841	0.5702
				0.0057	0.0076	0.0070	0.0073
skLinearSVC	[-2, -1, 0]	571.8	0.27	0.7143	0.5516	0.5747	0.5629
				0.0060	0.0078	0.0078	0.0077
skLinearSVC	[-3, -2, -1, 0]	486.9	0.37	0.7069	0.5383	0.5590	0.5485
				0.0073	0.0102	0.0097	0.0099
skLinearSVC	[0, 1]	930.1	0.21	0.7195	0.5567	0.5845	0.5703
				0.0057	0.0073	0.0075	0.0073
skLinearSVC	[0, 1, 2]	545.6	0.25	0.7131	0.5515	0.5758	0.5634
				0.0070	0.0085	0.0092	0.0088
skLinearSVC	[0, 1, 2, 3]	453.2	0.41	0.7070	0.5419	0.5641	0.5528
				0.0056	0.0069	0.0079	0.0073

Table 7.9: Results of the different Part of Speech-Taggers on the Language Czech for the MSD Tagset

Tagger	Configuration	Time (seconds) Training	Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	26.2	7.82	0.8296 0.0047	0.6705 0.0074	0.6726 0.0076	0.6716 0.0075
nltkBrill	brill24	19.9	7.73	0.8287 0.0046	0.6693 0.0074	0.6720 0.0074	0.6706 0.0074
nltkBrill	nltkdemo18	12.5	7.58	0.8279 0.0046	0.6679 0.0073	0.6714 0.0074	0.6697 0.0073
nltkBrill	nltkdemo18plus	18.5	7.53	0.8286 0.0046	0.6691 0.0071	0.6719 0.0073	0.6705 0.0072
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	5.8	0.26	0.8441 0.0036	0.7531 0.0055	0.7191 0.0058	0.7357 0.0055
skMultinomialNB	[-2, -1, 0, 1, 2]	3.9	0.36	0.8590 0.0025	0.7712 0.0054	0.7443 0.0051	0.7575 0.0051
skMultinomialNB	[-1, 0, 1]	2.7	0.20	0.8742 0.0033	0.7882 0.0046	0.7712 0.0056	0.7796 0.0050
skMultinomialNB	[0]	1.1	0.09	0.8726 0.0031	0.7613 0.0043	0.7665 0.0045	0.7639 0.0044
skMultinomialNB	[-1, 0]	2.0	0.14	0.8704 0.0019	0.7746 0.0046	0.7643 0.0037	0.7694 0.0041
skMultinomialNB	[-2, -1, 0]	2.7	0.16	0.8543 0.0027	0.7524 0.0063	0.7348 0.0057	0.7435 0.0059
skMultinomialNB	[-3, -2, -1, 0]	3.3	0.19	0.8425 0.0035	0.7363 0.0050	0.7141 0.0058	0.7250 0.0053
skMultinomialNB	[0, 1]	1.9	0.13	0.8659 0.0036	0.7638 0.0053	0.7522 0.0059	0.7580 0.0055
skMultinomialNB	[0, 1, 2]	2.7	0.14	0.8553 0.0031	0.7497 0.0053	0.7319 0.0054	0.7407 0.0053
skMultinomialNB	[0, 1, 2, 3]	3.5	0.14	0.8444 0.0028	0.7347 0.0049	0.7124 0.0054	0.7234 0.0051
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	17.5	0.18	0.8286 0.0055	0.7012 0.0066	0.6795 0.0078	0.6902 0.0072
skPerceptron	[-2, -1, 0, 1, 2]	14.4	0.12	0.8388 0.0053	0.7167 0.0071	0.6983 0.0083	0.7074 0.0076
skPerceptron	[-1, 0, 1]	10.7	0.10	0.8521 0.0053	0.7388 0.0099	0.7264 0.0103	0.7326 0.0101
skPerceptron	[0]	6.2	0.04	0.8315 0.0165	0.6802 0.0320	0.6848 0.0324	0.6825 0.0322
skPerceptron	[-1, 0]	8.8	0.08	0.8283 0.0069	0.7011 0.0129	0.6873 0.0133	0.6941 0.0131
skPerceptron	[-2, -1, 0]	10.9	0.08	0.8247 0.0075	0.6909 0.0123	0.6759 0.0141	0.6833 0.0132
skPerceptron	[-3, -2, -1, 0]	13.0	0.13	0.8255 0.0071	0.6929 0.0130	0.6772 0.0143	0.6849 0.0136
skPerceptron	[0, 1]	8.4	0.06	0.8394 0.0066	0.7072 0.0108	0.6982 0.0122	0.7026 0.0114
skPerceptron	[0, 1, 2]	10.9	0.12	0.8528 0.0064	0.7358 0.0108	0.7252 0.0109	0.7305 0.0108
skPerceptron	[0, 1, 2, 3]	13.0	0.17	0.8466 0.0041	0.7262 0.0075	0.7133 0.0074	0.7197 0.0074
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	24.2	0.22	0.8520 0.0041	0.7414 0.0060	0.7263 0.0065	0.7338 0.0062
skLinearSVC	[-2, -1, 0, 1, 2]	19.8	0.13	0.8679 0.0044	0.7655 0.0064	0.7552 0.0072	0.7603 0.0068
skLinearSVC	[-1, 0, 1]	14.1	0.09	0.8791 0.0031	0.7808 0.0049	0.7759 0.0056	0.7783 0.0052
skLinearSVC	[0]	21.3	0.08	0.8801 0.0032	0.7731 0.0041	0.7783 0.0043	0.7757 0.0042
skLinearSVC	[-1, 0]	15.6	0.08	0.8797 0.0043	0.7783 0.0074	0.7783 0.0073	0.7783 0.0073
skLinearSVC	[-2, -1, 0]	18.6	0.14	0.8754 0.0047	0.7726 0.0076	0.7697 0.0082	0.7712 0.0079
skLinearSVC	[-3, -2, -1, 0]	20.1	0.09	0.8681 0.0048	0.7606 0.0077	0.7552 0.0080	0.7579 0.0079
skLinearSVC	[0, 1]	14.6	0.05	0.8908 0.0031	0.7979 0.0053	0.7983 0.0059	0.7981 0.0055
skLinearSVC	[0, 1, 2]	16.4	0.10	0.8871 0.0036	0.7918 0.0062	0.7907 0.0067	0.7912 0.0065
skLinearSVC	[0, 1, 2, 3]	17.7	0.10	0.8825 0.0030	0.7844 0.0053	0.7820 0.0056	0.7832 0.0054

Table 7.10: Results of the different Part of Speech-Taggers on the Language Czech for the Universal Tagset

Tagger	Configuration	Time (seconds) Training	Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	69.9	8.74	0.7793	0.6126	0.6290	0.6207
				0.0066	0.0100	0.0109	0.0104
nltkBrill	brill24	67.1	7.45	0.7782	0.6117	0.6283	0.6199
				0.0068	0.0103	0.0108	0.0105
nltkBrill	nltkdemo18	52.6	7.98	0.7776	0.6117	0.6278	0.6197
				0.0069	0.0103	0.0110	0.0106
nltkBrill	nltkdemo18plus	57.3	7.94	0.7776	0.6115	0.6277	0.6195
				0.0067	0.0103	0.0108	0.0105
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	11.3	0.71	0.5094	0.3543	0.3204	0.3365
				0.0076	0.0105	0.0083	0.0093
skMultinomialNB	[-2, -1, 0, 1, 2]	8.6	0.58	0.5471	0.3804	0.3464	0.3626
				0.0071	0.0090	0.0073	0.0080
skMultinomialNB	[-1, 0, 1]	5.6	0.42	0.5907	0.4086	0.3791	0.3933
				0.0062	0.0077	0.0067	0.0071
skMultinomialNB	[0]	2.5	0.21	0.5846	0.3314	0.3487	0.3398
				0.0074	0.0089	0.0094	0.0091
skMultinomialNB	[-1, 0]	4.1	0.33	0.5886	0.3920	0.3754	0.3836
				0.0070	0.0079	0.0067	0.0072
skMultinomialNB	[-2, -1, 0]	5.6	0.37	0.5612	0.3772	0.3531	0.3648
				0.0074	0.0089	0.0075	0.0081
skMultinomialNB	[-3, -2, -1, 0]	6.7	0.55	0.5363	0.3623	0.3359	0.3486
				0.0072	0.0082	0.0066	0.0073
skMultinomialNB	[0, 1]	4.0	0.31	0.5842	0.3726	0.3559	0.3641
				0.0049	0.0076	0.0067	0.0070
skMultinomialNB	[0, 1, 2]	5.4	0.39	0.5572	0.3582	0.3325	0.3448
				0.0050	0.0067	0.0058	0.0061
skMultinomialNB	[0, 1, 2, 3]	7.0	0.49	0.5321	0.3430	0.3141	0.3279
				0.0057	0.0077	0.0064	0.0069
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	491.3	0.61	0.7041	0.5209	0.5132	0.5170
				0.0067	0.0101	0.0086	0.0093
skPerceptron	[-2, -1, 0, 1, 2]	462.7	0.52	0.7152	0.5460	0.5415	0.5437
				0.0055	0.0083	0.0072	0.0076
skPerceptron	[-1, 0, 1]	382.5	0.34	0.7124	0.5621	0.5599	0.5610
				0.0064	0.0078	0.0075	0.0075
skPerceptron	[0]	213.9	0.14	0.7277	0.5809	0.6112	0.5956
				0.0121	0.0112	0.0122	0.0117
skPerceptron	[-1, 0]	315.4	0.22	0.7162	0.5707	0.5735	0.5721
				0.0089	0.0091	0.0102	0.0095
skPerceptron	[-2, -1, 0]	388.4	0.35	0.7155	0.5693	0.5674	0.5683
				0.0087	0.0107	0.0097	0.0101
skPerceptron	[-3, -2, -1, 0]	463.2	0.40	0.7120	0.5593	0.5556	0.5575
				0.0099	0.0103	0.0099	0.0100
skPerceptron	[0, 1]	344.0	0.23	0.7052	0.5662	0.5627	0.5644
				0.0097	0.0099	0.0101	0.0099
skPerceptron	[0, 1, 2]	407.3	0.32	0.7078	0.5572	0.5469	0.5520
				0.0067	0.0100	0.0088	0.0093
skPerceptron	[0, 1, 2, 3]	462.6	0.46	0.7010	0.5440	0.5316	0.5377
				0.0040	0.0086	0.0078	0.0080
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	161.9	0.27	0.7273	0.5604	0.5615	0.5609
				0.0059	0.0085	0.0071	0.0077
skLinearSVC	[-2, -1, 0, 1, 2]	159.2	0.26	0.7443	0.5871	0.5920	0.5896
				0.0066	0.0106	0.0102	0.0104
skLinearSVC	[-1, 0, 1]	202.1	0.18	0.7663	0.6251	0.6362	0.6306
				0.0062	0.0092	0.0092	0.0091
skLinearSVC	[0]	614.4	0.08	0.7703	0.6239	0.6564	0.6397
				0.0065	0.0095	0.0103	0.0099
skLinearSVC	[-1, 0]	487.6	0.11	0.7676	0.6302	0.6472	0.6385
				0.0064	0.0100	0.0096	0.0098
skLinearSVC	[-2, -1, 0]	271.7	0.19	0.7656	0.6288	0.6433	0.6359
				0.0067	0.0104	0.0096	0.0100
skLinearSVC	[-3, -2, -1, 0]	224.4	0.21	0.7600	0.6187	0.6304	0.6245
				0.0061	0.0096	0.0088	0.0092
skLinearSVC	[0, 1]	435.1	0.11	0.7714	0.6349	0.6532	0.6439
				0.0063	0.0085	0.0090	0.0087
skLinearSVC	[0, 1, 2]	235.1	0.16	0.7669	0.6286	0.6448	0.6366
				0.0061	0.0086	0.0091	0.0088
skLinearSVC	[0, 1, 2, 3]	199.2	0.21	0.7600	0.6169	0.6321	0.6244
				0.0064	0.0095	0.0091	0.0092

Table 7.11: Results of the different Part of Speech-Taggers on the Language Estonian for the MSD Tagset

Tagger	Configuration	Time (seconds) Training	Time (seconds) Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	32.4	7.57	0.8209 0.0068	0.6405 0.0112	0.6464 0.0107	0.6434 0.0109
nltkBrill	brill24	28.6	7.19	0.8195 0.0063	0.6381 0.0107	0.6455 0.0105	0.6418 0.0106
nltkBrill	nltkdemo18	18.0	6.98	0.8189 0.0062	0.6375 0.0108	0.6450 0.0105	0.6412 0.0106
nltkBrill	nltkdemo18plus	23.8	7.60	0.8194 0.0064	0.6380 0.0109	0.6453 0.0106	0.6416 0.0107
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	5.5	0.22	0.8250 0.0052	0.7129 0.0045	0.6802 0.0064	0.6961 0.0053
skMultinomialNB	[-2, -1, 0, 1, 2]	4.2	0.20	0.8401 0.0048	0.7301 0.0046	0.7033 0.0060	0.7164 0.0050
skMultinomialNB	[-1, 0, 1]	2.7	0.13	0.8614 0.0034	0.7558 0.0045	0.7409 0.0045	0.7483 0.0042
skMultinomialNB	[0]	1.0	0.09	0.8774 0.0022	0.7719 0.0041	0.7855 0.0045	0.7787 0.0043
skMultinomialNB	[-1, 0]	1.8	0.11	0.8677 0.0030	0.7622 0.0032	0.7549 0.0040	0.7586 0.0034
skMultinomialNB	[-2, -1, 0]	2.5	0.11	0.8474 0.0036	0.7305 0.0050	0.7157 0.0050	0.7230 0.0048
skMultinomialNB	[-3, -2, -1, 0]	3.3	0.18	0.8358 0.0037	0.7156 0.0048	0.6961 0.0045	0.7057 0.0044
skMultinomialNB	[0, 1]	1.8	0.12	0.8637 0.0024	0.7589 0.0063	0.7513 0.0056	0.7551 0.0058
skMultinomialNB	[0, 1, 2]	2.6	0.16	0.8460 0.0032	0.7329 0.0068	0.7167 0.0065	0.7247 0.0065
skMultinomialNB	[0, 1, 2, 3]	3.1	0.18	0.8360 0.0039	0.7196 0.0058	0.6990 0.0068	0.7091 0.0061
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	15.8	0.13	0.8324 0.0061	0.6901 0.0105	0.6756 0.0111	0.6827 0.0107
skPerceptron	[-2, -1, 0, 1, 2]	13.3	0.14	0.8514 0.0031	0.7222 0.0058	0.7120 0.0055	0.7171 0.0055
skPerceptron	[-1, 0, 1]	10.1	0.09	0.8721 0.0057	0.7612 0.0102	0.7552 0.0103	0.7582 0.0101
skPerceptron	[0]	5.6	0.03	0.8087 0.0067	0.6297 0.0109	0.6408 0.0112	0.6352 0.0110
skPerceptron	[-1, 0]	8.2	0.06	0.8564 0.0229	0.7332 0.0456	0.7282 0.0447	0.7307 0.0451
skPerceptron	[-2, -1, 0]	9.9	0.06	0.8460 0.0086	0.7141 0.0154	0.7058 0.0166	0.7099 0.0160
skPerceptron	[-3, -2, -1, 0]	11.9	0.11	0.8401 0.0061	0.7055 0.0113	0.6949 0.0096	0.7001 0.0104
skPerceptron	[0, 1]	8.4	0.06	0.8730 0.0064	0.7685 0.0088	0.7660 0.0099	0.7672 0.0092
skPerceptron	[0, 1, 2]	10.4	0.10	0.8704 0.0058	0.7636 0.0105	0.7586 0.0110	0.7611 0.0107
skPerceptron	[0, 1, 2, 3]	11.6	0.06	0.8648 0.0045	0.7564 0.0072	0.7482 0.0089	0.7523 0.0080
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	24.7	0.18	0.8841 0.0026	0.7840 0.0058	0.7793 0.0054	0.7817 0.0054
skLinearSVC	[-2, -1, 0, 1, 2]	19.2	0.16	0.8938 0.0030	0.8006 0.0061	0.7989 0.0058	0.7997 0.0058
skLinearSVC	[-1, 0, 1]	14.4	0.13	0.9067 0.0033	0.8234 0.0055	0.8256 0.0056	0.8245 0.0055
skLinearSVC	[0]	18.9	0.06	0.8996 0.0024	0.8070 0.0051	0.8212 0.0053	0.8141 0.0052
skLinearSVC	[-1, 0]	15.4	0.11	0.9067 0.0032	0.8242 0.0049	0.8299 0.0053	0.8271 0.0050
skLinearSVC	[-2, -1, 0]	17.0	0.11	0.9036 0.0033	0.8188 0.0054	0.8226 0.0055	0.8207 0.0054
skLinearSVC	[-3, -2, -1, 0]	20.6	0.12	0.8982 0.0039	0.8084 0.0066	0.8110 0.0065	0.8097 0.0065
skLinearSVC	[0, 1]	12.8	0.08	0.9026 0.0021	0.8144 0.0046	0.8231 0.0048	0.8187 0.0046
skLinearSVC	[0, 1, 2]	16.7	0.14	0.8968 0.0025	0.8039 0.0057	0.8106 0.0056	0.8072 0.0056
skLinearSVC	[0, 1, 2, 3]	19.2	0.14	0.8923 0.0022	0.7956 0.0042	0.8009 0.0043	0.7983 0.0041

Table 7.12: Results of the different Part of Speech-Taggers on the Language Estonian for the Universal Tagset

Tagger	Configuration	Time (seconds)		Accuracy	Recall	Precision	F1-Score
		Training	Tagging	(second line: standard deviation)			
nltkBrill	fntbl37	33.4	9.65	0.8077	0.6235	0.6282	0.6259
				0.0056	0.0120	0.0122	0.0121
nltkBrill	brill24	26.9	8.94	0.8076	0.6224	0.6277	0.6250
				0.0055	0.0122	0.0124	0.0123
nltkBrill	nltkdemo18	19.5	9.30	0.8064	0.6210	0.6270	0.6240
				0.0055	0.0120	0.0124	0.0122
nltkBrill	nltkdemo18plus	23.6	9.01	0.8067	0.6215	0.6273	0.6244
				0.0054	0.0120	0.0123	0.0122
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	15.1	1.25	0.5712	0.3544	0.3130	0.3324
				0.0038	0.0091	0.0072	0.0080
skMultinomialNB	[-2, -1, 0, 1, 2]	11.9	0.94	0.5956	0.3733	0.3344	0.3528
				0.0039	0.0090	0.0075	0.0081
skMultinomialNB	[-1, 0, 1]	7.8	0.66	0.6227	0.3907	0.3583	0.3738
				0.0057	0.0105	0.0094	0.0099
skMultinomialNB	[0]	3.6	0.31	0.6429	0.3725	0.3775	0.3750
				0.0065	0.0102	0.0104	0.0103
skMultinomialNB	[-1, 0]	5.6	0.48	0.6249	0.3774	0.3533	0.3649
				0.0063	0.0091	0.0088	0.0089
skMultinomialNB	[-2, -1, 0]	7.7	0.55	0.6058	0.3638	0.3322	0.3473
				0.0057	0.0089	0.0083	0.0086
skMultinomialNB	[-3, -2, -1, 0]	9.6	0.77	0.5891	0.3505	0.3156	0.3321
				0.0050	0.0080	0.0071	0.0074
skMultinomialNB	[0, 1]	5.7	0.48	0.6284	0.3815	0.3565	0.3686
				0.0049	0.0087	0.0081	0.0083
skMultinomialNB	[0, 1, 2]	7.9	0.63	0.6086	0.3688	0.3367	0.3520
				0.0040	0.0087	0.0078	0.0082
skMultinomialNB	[0, 1, 2, 3]	9.7	0.86	0.5921	0.3567	0.3211	0.3380
				0.0048	0.0103	0.0089	0.0095
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	806.4	0.92	0.7413	0.5182	0.5034	0.5107
				0.0064	0.0125	0.0131	0.0128
skPerceptron	[-2, -1, 0, 1, 2]	670.5	0.76	0.7605	0.5536	0.5417	0.5476
				0.0064	0.0125	0.0128	0.0126
skPerceptron	[-1, 0, 1]	551.2	0.45	0.7798	0.5924	0.5843	0.5883
				0.0055	0.0115	0.0122	0.0118
skPerceptron	[0]	339.2	0.22	0.7968	0.6180	0.6264	0.6222
				0.0038	0.0122	0.0127	0.0124
skPerceptron	[-1, 0]	450.2	0.37	0.7890	0.6095	0.6063	0.6079
				0.0055	0.0110	0.0116	0.0113
skPerceptron	[-2, -1, 0]	549.8	0.48	0.7828	0.5961	0.5904	0.5932
				0.0046	0.0096	0.0101	0.0098
skPerceptron	[-3, -2, -1, 0]	613.3	0.63	0.7733	0.5789	0.5718	0.5753
				0.0041	0.0094	0.0096	0.0095
skPerceptron	[0, 1]	439.7	0.33	0.7626	0.5993	0.5915	0.5954
				0.0084	0.0120	0.0121	0.0120
skPerceptron	[0, 1, 2]	530.9	0.49	0.7558	0.5925	0.5801	0.5862
				0.0041	0.0118	0.0123	0.0120
skPerceptron	[0, 1, 2, 3]	589.8	0.59	0.7490	0.5803	0.5635	0.5718
				0.0050	0.0112	0.0116	0.0114
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	231.5	0.36	0.7934	0.6063	0.6009	0.6036
				0.0059	0.0127	0.0131	0.0129
skLinearSVC	[-2, -1, 0, 1, 2]	233.3	0.27	0.8122	0.6394	0.6371	0.6383
				0.0063	0.0127	0.0133	0.0130
skLinearSVC	[-1, 0, 1]	339.5	0.28	0.8262	0.6642	0.6644	0.6643
				0.0052	0.0119	0.0120	0.0120
skLinearSVC	[0]	653.8	0.13	0.8194	0.6526	0.6615	0.6570
				0.0048	0.0109	0.0112	0.0110
skLinearSVC	[-1, 0]	795.7	0.15	0.8183	0.6534	0.6565	0.6549
				0.0047	0.0102	0.0104	0.0103
skLinearSVC	[-2, -1, 0]	405.3	0.23	0.8143	0.6461	0.6481	0.6471
				0.0059	0.0126	0.0130	0.0128
skLinearSVC	[-3, -2, -1, 0]	325.0	0.19	0.8106	0.6392	0.6399	0.6396
				0.0060	0.0129	0.0132	0.0130
skLinearSVC	[0, 1]	747.7	0.17	0.8269	0.6641	0.6662	0.6652
				0.0045	0.0102	0.0105	0.0103
skLinearSVC	[0, 1, 2]	413.3	0.25	0.8258	0.6638	0.6647	0.6642
				0.0053	0.0105	0.0110	0.0107
skLinearSVC	[0, 1, 2, 3]	329.7	0.24	0.8220	0.6572	0.6572	0.6572
				0.0053	0.0112	0.0113	0.0112

Table 7.13: Results of the different Part of Speech-Taggers on the Language Hungarian for the MSD Tagset

Tagger	Configuration	Time (seconds) Training	Time (seconds) Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	35.6	8.25	0.8093	0.6259	0.6295	0.6277
				0.0052	0.0118	0.0119	0.0118
nltkBrill	brill24	28.4	8.00	0.8089	0.6243	0.6287	0.6265
				0.0054	0.0119	0.0120	0.0119
nltkBrill	nltkdemo18	18.0	7.89	0.8077	0.6226	0.6276	0.6251
				0.0053	0.0115	0.0119	0.0117
nltkBrill	nltkdemo18plus	24.0	8.15	0.8084	0.6241	0.6283	0.6262
				0.0053	0.0115	0.0119	0.0117
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	5.9	0.27	0.8456	0.7631	0.7285	0.7454
				0.0026	0.0060	0.0060	0.0059
skMultinomialNB	[-2, -1, 0, 1, 2]	4.5	0.21	0.8579	0.7750	0.7467	0.7606
				0.0032	0.0066	0.0073	0.0069
skMultinomialNB	[-1, 0, 1]	2.8	0.19	0.8736	0.7916	0.7724	0.7819
				0.0032	0.0078	0.0075	0.0076
skMultinomialNB	[0]	1.1	0.08	0.8763	0.7750	0.7842	0.7796
				0.0040	0.0084	0.0091	0.0087
skMultinomialNB	[-1, 0]	1.9	0.11	0.8631	0.7657	0.7568	0.7612
				0.0046	0.0103	0.0102	0.0102
skMultinomialNB	[-2, -1, 0]	2.7	0.13	0.8498	0.7505	0.7342	0.7422
				0.0060	0.0124	0.0124	0.0124
skMultinomialNB	[-3, -2, -1, 0]	3.4	0.15	0.8412	0.7424	0.7209	0.7315
				0.0053	0.0114	0.0115	0.0114
skMultinomialNB	[0, 1]	2.0	0.11	0.8689	0.7753	0.7635	0.7693
				0.0028	0.0062	0.0069	0.0065
skMultinomialNB	[0, 1, 2]	2.7	0.16	0.8535	0.7543	0.7358	0.7449
				0.0028	0.0053	0.0067	0.0060
skMultinomialNB	[0, 1, 2, 3]	3.5	0.16	0.8440	0.7416	0.7188	0.7300
				0.0026	0.0053	0.0061	0.0056
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	16.7	0.15	0.8353	0.7084	0.6920	0.7001
				0.0039	0.0069	0.0078	0.0073
skPerceptron	[-2, -1, 0, 1, 2]	14.4	0.14	0.8484	0.7313	0.7183	0.7247
				0.0061	0.0140	0.0137	0.0138
skPerceptron	[-1, 0, 1]	10.5	0.08	0.8587	0.7543	0.7446	0.7495
				0.0050	0.0124	0.0115	0.0119
skPerceptron	[0]	6.3	0.05	0.7981	0.6192	0.6265	0.6228
				0.0046	0.0121	0.0125	0.0123
skPerceptron	[-1, 0]	8.8	0.07	0.8163	0.6798	0.6700	0.6749
				0.0130	0.0204	0.0220	0.0212
skPerceptron	[-2, -1, 0]	10.7	0.08	0.8320	0.7027	0.6904	0.6965
				0.0051	0.0115	0.0121	0.0117
skPerceptron	[-3, -2, -1, 0]	12.4	0.07	0.8377	0.7098	0.7004	0.7051
				0.0049	0.0114	0.0109	0.0111
skPerceptron	[0, 1]	8.6	0.08	0.8297	0.7428	0.7370	0.7399
				0.0154	0.0209	0.0219	0.0214
skPerceptron	[0, 1, 2]	10.8	0.07	0.8339	0.7390	0.7311	0.7350
				0.0095	0.0123	0.0126	0.0124
skPerceptron	[0, 1, 2, 3]	12.4	0.08	0.8370	0.7439	0.7345	0.7391
				0.0061	0.0122	0.0120	0.0121
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	23.5	0.21	0.8715	0.7720	0.7628	0.7674
				0.0041	0.0078	0.0082	0.0080
skLinearSVC	[-2, -1, 0, 1, 2]	18.0	0.15	0.8845	0.7986	0.7930	0.7958
				0.0050	0.0090	0.0093	0.0091
skLinearSVC	[-1, 0, 1]	15.4	0.16	0.8882	0.8173	0.8153	0.8163
				0.0033	0.0064	0.0068	0.0066
skLinearSVC	[0]	19.5	0.09	0.8869	0.7912	0.8006	0.7959
				0.0039	0.0089	0.0095	0.0092
skLinearSVC	[-1, 0]	16.7	0.17	0.8985	0.8171	0.8206	0.8188
				0.0040	0.0092	0.0095	0.0093
skLinearSVC	[-2, -1, 0]	18.8	0.11	0.8933	0.8077	0.8096	0.8086
				0.0039	0.0093	0.0091	0.0092
skLinearSVC	[-3, -2, -1, 0]	20.4	0.12	0.8870	0.7964	0.7968	0.7966
				0.0042	0.0094	0.0101	0.0097
skLinearSVC	[0, 1]	14.6	0.05	0.8744	0.8062	0.8083	0.8073
				0.0035	0.0067	0.0079	0.0073
skLinearSVC	[0, 1, 2]	17.6	0.13	0.8748	0.8018	0.8016	0.8017
				0.0051	0.0087	0.0096	0.0091
skLinearSVC	[0, 1, 2, 3]	19.9	0.16	0.8703	0.7873	0.7850	0.7861
				0.0044	0.0081	0.0087	0.0084

Table 7.14: Results of the different Part of Speech-Taggers on the Language Hungarian for the Universal Tagset

Tagger	Configuration	Time (seconds) Training	Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	95.7	10.54	0.7170	0.5491	0.5757	0.5621
				0.0054	0.0077	0.0085	0.0081
nltkBrill	brill24	77.5	10.17	0.7173	0.5500	0.5764	0.5629
				0.0055	0.0075	0.0083	0.0078
nltkBrill	nltkdemo18	62.8	10.31	0.7169	0.5497	0.5758	0.5624
				0.0053	0.0076	0.0084	0.0079
nltkBrill	nltkdemo18plus	70.0	10.55	0.7169	0.5497	0.5758	0.5624
				0.0053	0.0076	0.0084	0.0079
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	21.4	1.78	0.4203	0.2181	0.1966	0.2068
				0.0069	0.0062	0.0061	0.0061
skMultinomialNB	[-2, -1, 0, 1, 2]	15.2	1.47	0.4521	0.2464	0.2236	0.2344
				0.0058	0.0053	0.0051	0.0052
skMultinomialNB	[-1, 0, 1]	9.7	0.96	0.4899	0.2764	0.2566	0.2662
				0.0072	0.0069	0.0063	0.0065
skMultinomialNB	[0]	4.4	0.42	0.4522	0.2005	0.2156	0.2077
				0.0027	0.0054	0.0057	0.0055
skMultinomialNB	[-1, 0]	7.4	0.69	0.4855	0.2706	0.2590	0.2647
				0.0062	0.0070	0.0068	0.0069
skMultinomialNB	[-2, -1, 0]	10.1	0.84	0.4661	0.2583	0.2412	0.2495
				0.0066	0.0074	0.0074	0.0074
skMultinomialNB	[-3, -2, -1, 0]	12.8	1.03	0.4478	0.2427	0.2243	0.2332
				0.0061	0.0076	0.0078	0.0077
skMultinomialNB	[0, 1]	7.3	0.66	0.4606	0.2276	0.2205	0.2240
				0.0047	0.0041	0.0044	0.0042
skMultinomialNB	[0, 1, 2]	10.2	0.96	0.4366	0.2115	0.1998	0.2055
				0.0053	0.0052	0.0052	0.0052
skMultinomialNB	[0, 1, 2, 3]	12.5	1.04	0.4182	0.1979	0.1854	0.1914
				0.0052	0.0055	0.0054	0.0054
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	1164.2	1.74	0.6170	0.4259	0.4245	0.4252
				0.0048	0.0058	0.0063	0.0060
skPerceptron	[-2, -1, 0, 1, 2]	989.4	1.12	0.6344	0.4535	0.4553	0.4544
				0.0054	0.0079	0.0080	0.0079
skPerceptron	[-1, 0, 1]	771.4	0.67	0.6554	0.4859	0.4924	0.4891
				0.0071	0.0073	0.0076	0.0074
skPerceptron	[0]	417.7	0.32	0.6523	0.5054	0.5436	0.5238
				0.0104	0.0082	0.0090	0.0086
skPerceptron	[-1, 0]	623.3	0.52	0.6424	0.4885	0.4949	0.4917
				0.0084	0.0091	0.0100	0.0095
skPerceptron	[-2, -1, 0]	759.0	0.64	0.6446	0.4791	0.4843	0.4817
				0.0058	0.0078	0.0080	0.0079
skPerceptron	[-3, -2, -1, 0]	853.7	0.87	0.6425	0.4723	0.4776	0.4749
				0.0064	0.0090	0.0101	0.0095
skPerceptron	[0, 1]	631.6	0.52	0.6421	0.4958	0.5034	0.4996
				0.0065	0.0065	0.0078	0.0071
skPerceptron	[0, 1, 2]	763.1	0.65	0.6377	0.4797	0.4792	0.4794
				0.0058	0.0062	0.0070	0.0065
skPerceptron	[0, 1, 2, 3]	858.6	0.90	0.6245	0.4633	0.4589	0.4611
				0.0059	0.0049	0.0072	0.0060
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	301.5	0.38	0.6596	0.4835	0.4920	0.4877
				0.0047	0.0065	0.0073	0.0068
skLinearSVC	[-2, -1, 0, 1, 2]	300.6	0.32	0.6827	0.5200	0.5325	0.5262
				0.0046	0.0065	0.0068	0.0066
skLinearSVC	[-1, 0, 1]	368.6	0.32	0.7045	0.5575	0.5753	0.5663
				0.0042	0.0077	0.0085	0.0080
skLinearSVC	[0]	1348.0	0.16	0.6864	0.5270	0.5668	0.5462
				0.0041	0.0084	0.0089	0.0086
skLinearSVC	[-1, 0]	866.9	0.26	0.6991	0.5564	0.5776	0.5668
				0.0050	0.0079	0.0091	0.0085
skLinearSVC	[-2, -1, 0]	466.9	0.34	0.6937	0.5488	0.5676	0.5581
				0.0055	0.0082	0.0089	0.0085
skLinearSVC	[-3, -2, -1, 0]	397.7	0.30	0.6844	0.5338	0.5507	0.5421
				0.0054	0.0075	0.0088	0.0081
skLinearSVC	[0, 1]	868.6	0.20	0.6985	0.5505	0.5745	0.5622
				0.0045	0.0079	0.0086	0.0082
skLinearSVC	[0, 1, 2]	457.0	0.31	0.6941	0.5463	0.5674	0.5567
				0.0054	0.0082	0.0091	0.0086
skLinearSVC	[0, 1, 2, 3]	388.7	0.24	0.6844	0.5352	0.5561	0.5454
				0.0055	0.0081	0.0086	0.0083

Table 7.15: Results of the different Part of Speech-Taggers on the Language Polish for the MSD Tagset

Tagger	Configuration	Time (seconds) Training	Time (seconds) Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	13.4	8.07	0.8167	0.6546	0.6563	0.6555
				0.0046	0.0095	0.0094	0.0095
nltkBrill	brill24	10.2	7.55	0.8162	0.6536	0.6558	0.6547
				0.0049	0.0097	0.0096	0.0096
nltkBrill	nltkdemo18	7.2	7.71	0.8156	0.6530	0.6553	0.6541
				0.0049	0.0097	0.0097	0.0097
nltkBrill	nltkdemo18plus	9.4	7.65	0.8159	0.6538	0.6557	0.6547
				0.0049	0.0097	0.0097	0.0097
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	6.2	0.28	0.8280	0.7355	0.6986	0.7166
				0.0026	0.0043	0.0040	0.0039
skMultinomialNB	[-2, -1, 0, 1, 2]	4.6	0.18	0.8446	0.7526	0.7239	0.7379
				0.0022	0.0023	0.0027	0.0022
skMultinomialNB	[-1, 0, 1]	2.9	0.17	0.8645	0.7745	0.7555	0.7649
				0.0027	0.0035	0.0040	0.0037
skMultinomialNB	[0]	1.2	0.10	0.8670	0.7545	0.7585	0.7565
				0.0029	0.0061	0.0060	0.0061
skMultinomialNB	[-1, 0]	2.0	0.16	0.8623	0.7620	0.7506	0.7562
				0.0027	0.0045	0.0045	0.0044
skMultinomialNB	[-2, -1, 0]	2.9	0.15	0.8448	0.7383	0.7200	0.7291
				0.0027	0.0053	0.0047	0.0049
skMultinomialNB	[-3, -2, -1, 0]	3.7	0.13	0.8330	0.7239	0.7011	0.7123
				0.0033	0.0052	0.0062	0.0056
skMultinomialNB	[0, 1]	2.0	0.12	0.8622	0.7609	0.7496	0.7552
				0.0024	0.0048	0.0043	0.0045
skMultinomialNB	[0, 1, 2]	2.8	0.16	0.8463	0.7406	0.7221	0.7312
				0.0034	0.0047	0.0052	0.0049
skMultinomialNB	[0, 1, 2, 3]	3.4	0.19	0.8341	0.7255	0.7023	0.7137
				0.0034	0.0050	0.0052	0.0050
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	18.3	0.13	0.8382	0.7164	0.6998	0.7080
				0.0057	0.0065	0.0070	0.0067
skPerceptron	[-2, -1, 0, 1, 2]	15.6	0.12	0.8501	0.7352	0.7216	0.7283
				0.0067	0.0107	0.0113	0.0110
skPerceptron	[-1, 0, 1]	12.5	0.08	0.8738	0.7782	0.7685	0.7733
				0.0071	0.0111	0.0117	0.0114
skPerceptron	[0]	6.8	0.03	0.8125	0.6537	0.6571	0.6554
				0.0073	0.0100	0.0099	0.0100
skPerceptron	[-1, 0]	9.8	0.06	0.8284	0.6919	0.6836	0.6877
				0.0076	0.0157	0.0161	0.0159
skPerceptron	[-2, -1, 0]	12.2	0.08	0.8342	0.7058	0.6943	0.7000
				0.0084	0.0137	0.0143	0.0140
skPerceptron	[-3, -2, -1, 0]	14.0	0.09	0.8292	0.6974	0.6833	0.6903
				0.0059	0.0103	0.0097	0.0099
skPerceptron	[0, 1]	9.5	0.05	0.8647	0.7670	0.7596	0.7633
				0.0101	0.0214	0.0212	0.0213
skPerceptron	[0, 1, 2]	12.2	0.09	0.8654	0.7630	0.7538	0.7583
				0.0092	0.0149	0.0159	0.0154
skPerceptron	[0, 1, 2, 3]	13.9	0.09	0.8643	0.7609	0.7504	0.7556
				0.0028	0.0081	0.0078	0.0079
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	26.2	0.23	0.8733	0.7762	0.7663	0.7712
				0.0027	0.0033	0.0038	0.0035
skLinearSVC	[-2, -1, 0, 1, 2]	21.7	0.13	0.8881	0.8001	0.7938	0.7969
				0.0031	0.0039	0.0042	0.0040
skLinearSVC	[-1, 0, 1]	17.7	0.13	0.9042	0.8268	0.8237	0.8252
				0.0029	0.0046	0.0048	0.0047
skLinearSVC	[0]	20.7	0.05	0.8796	0.7749	0.7790	0.7770
				0.0028	0.0062	0.0060	0.0061
skLinearSVC	[-1, 0]	19.0	0.08	0.8820	0.7830	0.7825	0.7828
				0.0045	0.0077	0.0077	0.0077
skLinearSVC	[-2, -1, 0]	21.6	0.13	0.8744	0.7716	0.7684	0.7700
				0.0042	0.0063	0.0064	0.0063
skLinearSVC	[-3, -2, -1, 0]	24.3	0.14	0.8656	0.7579	0.7522	0.7551
				0.0040	0.0047	0.0055	0.0050
skLinearSVC	[0, 1]	19.1	0.12	0.8960	0.8083	0.8093	0.8088
				0.0026	0.0044	0.0037	0.0040
skLinearSVC	[0, 1, 2]	18.5	0.08	0.8905	0.7992	0.7986	0.7989
				0.0029	0.0049	0.0047	0.0048
skLinearSVC	[0, 1, 2, 3]	22.2	0.13	0.8829	0.7862	0.7846	0.7854
				0.0035	0.0053	0.0052	0.0052

Table 7.16: Results of the different Part of Speech-Taggers on the Language Polish for the Universal Tagset

Tagger	Configuration	Time (seconds) Training	Time (seconds) Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	112.0	10.38	0.7264	0.5531	0.5842	0.5682
				0.0061	0.0071	0.0080	0.0075
nltkBrill	brill24	91.5	9.97	0.7260	0.5525	0.5839	0.5678
				0.0062	0.0074	0.0083	0.0078
nltkBrill	nltkdemo18	72.9	10.03	0.7252	0.5521	0.5836	0.5674
				0.0061	0.0075	0.0085	0.0080
nltkBrill	nltkdemo18plus	81.3	10.35	0.7253	0.5522	0.5836	0.5675
				0.0060	0.0076	0.0085	0.0080
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	24.0	1.90	0.4333	0.2333	0.2071	0.2194
				0.0069	0.0095	0.0081	0.0087
skMultinomialNB	[-2, -1, 0, 1, 2]	18.5	1.52	0.4644	0.2589	0.2317	0.2445
				0.0061	0.0083	0.0074	0.0078
skMultinomialNB	[-1, 0, 1]	12.0	0.99	0.5006	0.2859	0.2628	0.2739
				0.0070	0.0081	0.0075	0.0077
skMultinomialNB	[0]	5.2	0.50	0.4974	0.2361	0.2564	0.2458
				0.0075	0.0082	0.0090	0.0086
skMultinomialNB	[-1, 0]	8.5	0.93	0.5072	0.2894	0.2764	0.2827
				0.0056	0.0068	0.0061	0.0064
skMultinomialNB	[-2, -1, 0]	11.6	1.25	0.4868	0.2775	0.2566	0.2666
				0.0051	0.0069	0.0059	0.0063
skMultinomialNB	[-3, -2, -1, 0]	14.3	1.26	0.4639	0.2585	0.2357	0.2466
				0.0056	0.0056	0.0052	0.0053
skMultinomialNB	[0, 1]	8.6	0.78	0.4851	0.2479	0.2355	0.2415
				0.0070	0.0085	0.0082	0.0083
skMultinomialNB	[0, 1, 2]	11.6	0.88	0.4589	0.2301	0.2108	0.2200
				0.0070	0.0084	0.0078	0.0081
skMultinomialNB	[0, 1, 2, 3]	15.0	1.35	0.4398	0.2170	0.1961	0.2060
				0.0058	0.0079	0.0069	0.0073
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	1426.8	1.96	0.6351	0.4298	0.4282	0.4290
				0.0064	0.0067	0.0066	0.0065
skPerceptron	[-2, -1, 0, 1, 2]	1258.0	1.30	0.6478	0.4515	0.4530	0.4523
				0.0071	0.0083	0.0081	0.0082
skPerceptron	[-1, 0, 1]	1009.3	0.93	0.6639	0.4843	0.4889	0.4866
				0.0070	0.0077	0.0079	0.0078
skPerceptron	[0]	583.9	0.42	0.6759	0.5133	0.5573	0.5344
				0.0086	0.0082	0.0097	0.0089
skPerceptron	[-1, 0]	954.8	0.79	0.6497	0.4887	0.4955	0.4921
				0.0076	0.0079	0.0082	0.0079
skPerceptron	[-2, -1, 0]	1072.6	0.96	0.6504	0.4784	0.4835	0.4809
				0.0059	0.0083	0.0092	0.0087
skPerceptron	[-3, -2, -1, 0]	1181.3	1.18	0.6476	0.4711	0.4740	0.4726
				0.0052	0.0066	0.0066	0.0065
skPerceptron	[0, 1]	867.8	0.68	0.6632	0.5108	0.5158	0.5133
				0.0106	0.0085	0.0103	0.0094
skPerceptron	[0, 1, 2]	1038.8	0.84	0.6572	0.4948	0.4937	0.4943
				0.0069	0.0084	0.0089	0.0086
skPerceptron	[0, 1, 2, 3]	1133.4	1.06	0.6464	0.4757	0.4694	0.4725
				0.0065	0.0070	0.0070	0.0069
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	375.4	0.50	0.6718	0.4813	0.4879	0.4846
				0.0073	0.0077	0.0091	0.0084
skLinearSVC	[-2, -1, 0, 1, 2]	354.9	0.50	0.6925	0.5158	0.5270	0.5214
				0.0061	0.0073	0.0080	0.0076
skLinearSVC	[-1, 0, 1]	475.7	0.26	0.7169	0.5549	0.5738	0.5642
				0.0070	0.0074	0.0086	0.0079
skLinearSVC	[0]	1678.9	0.20	0.7032	0.5331	0.5788	0.5550
				0.0053	0.0074	0.0085	0.0079
skLinearSVC	[-1, 0]	1146.3	0.24	0.7131	0.5591	0.5854	0.5719
				0.0061	0.0081	0.0097	0.0088
skLinearSVC	[-2, -1, 0]	591.2	0.35	0.7074	0.5515	0.5730	0.5620
				0.0064	0.0080	0.0093	0.0086
skLinearSVC	[-3, -2, -1, 0]	486.3	0.33	0.6978	0.5348	0.5540	0.5442
				0.0056	0.0059	0.0073	0.0065
skLinearSVC	[0, 1]	987.1	0.24	0.7191	0.5636	0.5903	0.5767
				0.0073	0.0089	0.0100	0.0094
skLinearSVC	[0, 1, 2]	551.3	0.26	0.7144	0.5585	0.5820	0.5700
				0.0064	0.0082	0.0091	0.0086
skLinearSVC	[0, 1, 2, 3]	455.0	0.32	0.7062	0.5470	0.5686	0.5576
				0.0064	0.0073	0.0085	0.0078

Table 7.17: Results of the different Part of Speech-Taggers on the Language Slovak for the MSD Tagset

Tagger	Configuration	Time (seconds) Training	Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	53.1	8.54	0.8148	0.6491	0.6528	0.6509
				0.0050	0.0082	0.0079	0.0081
nltkBrill	brill24	41.9	8.00	0.8131	0.6469	0.6517	0.6493
				0.0052	0.0083	0.0079	0.0081
nltkBrill	nltkdemo18	28.8	7.71	0.8100	0.6444	0.6499	0.6471
				0.0051	0.0083	0.0080	0.0081
nltkBrill	nltkdemo18plus	39.1	8.35	0.8119	0.6462	0.6512	0.6487
				0.0053	0.0080	0.0078	0.0079
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	6.4	0.23	0.8196	0.7301	0.6919	0.7105
				0.0049	0.0087	0.0085	0.0083
skMultinomialNB	[-2, -1, 0, 1, 2]	4.8	0.25	0.8365	0.7471	0.7180	0.7323
				0.0035	0.0064	0.0067	0.0062
skMultinomialNB	[-1, 0, 1]	2.9	0.16	0.8573	0.7711	0.7538	0.7624
				0.0032	0.0058	0.0041	0.0047
skMultinomialNB	[0]	1.2	0.10	0.8518	0.7385	0.7497	0.7441
				0.0044	0.0070	0.0068	0.0069
skMultinomialNB	[-1, 0]	2.1	0.14	0.8499	0.7558	0.7469	0.7513
				0.0035	0.0065	0.0055	0.0059
skMultinomialNB	[-2, -1, 0]	3.0	0.15	0.8315	0.7304	0.7122	0.7212
				0.0033	0.0041	0.0046	0.0040
skMultinomialNB	[-3, -2, -1, 0]	3.7	0.19	0.8186	0.7148	0.6907	0.7025
				0.0038	0.0061	0.0065	0.0061
skMultinomialNB	[0, 1]	2.1	0.12	0.8444	0.7355	0.7266	0.7310
				0.0042	0.0080	0.0068	0.0073
skMultinomialNB	[0, 1, 2]	3.0	0.16	0.8332	0.7230	0.7067	0.7147
				0.0038	0.0062	0.0055	0.0057
skMultinomialNB	[0, 1, 2, 3]	3.8	0.17	0.8216	0.7085	0.6869	0.6976
				0.0040	0.0058	0.0048	0.0050
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	22.3	0.23	0.8152	0.6884	0.6686	0.6783
				0.0067	0.0103	0.0114	0.0108
skPerceptron	[-2, -1, 0, 1, 2]	16.9	0.14	0.8267	0.7075	0.6914	0.6994
				0.0076	0.0127	0.0132	0.0128
skPerceptron	[-1, 0, 1]	13.8	0.10	0.8398	0.7280	0.7177	0.7228
				0.0055	0.0088	0.0100	0.0094
skPerceptron	[0]	8.0	0.04	0.8057	0.6512	0.6611	0.6561
				0.0146	0.0332	0.0336	0.0334
skPerceptron	[-1, 0]	11.0	0.05	0.8230	0.6963	0.6913	0.6938
				0.0146	0.0297	0.0299	0.0298
skPerceptron	[-2, -1, 0]	13.2	0.09	0.8184	0.6917	0.6811	0.6863
				0.0083	0.0136	0.0132	0.0133
skPerceptron	[-3, -2, -1, 0]	16.5	0.15	0.8160	0.6906	0.6768	0.6836
				0.0084	0.0121	0.0119	0.0120
skPerceptron	[0, 1]	10.4	0.05	0.8567	0.7613	0.7594	0.7603
				0.0083	0.0127	0.0126	0.0126
skPerceptron	[0, 1, 2]	13.9	0.10	0.8521	0.7495	0.7432	0.7463
				0.0049	0.0097	0.0099	0.0097
skPerceptron	[0, 1, 2, 3]	15.4	0.11	0.8454	0.7384	0.7287	0.7335
				0.0053	0.0101	0.0105	0.0102
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	27.1	0.20	0.8346	0.7198	0.7077	0.7137
				0.0065	0.0105	0.0108	0.0106
skLinearSVC	[-2, -1, 0, 1, 2]	23.0	0.15	0.8515	0.7467	0.7392	0.7429
				0.0060	0.0108	0.0109	0.0108
skLinearSVC	[-1, 0, 1]	15.5	0.11	0.8686	0.7722	0.7709	0.7716
				0.0043	0.0089	0.0091	0.0089
skLinearSVC	[0]	23.0	0.07	0.8648	0.7570	0.7685	0.7627
				0.0040	0.0065	0.0063	0.0064
skLinearSVC	[-1, 0]	18.5	0.08	0.8625	0.7568	0.7611	0.7590
				0.0044	0.0082	0.0080	0.0081
skLinearSVC	[-2, -1, 0]	19.9	0.15	0.8574	0.7511	0.7519	0.7515
				0.0057	0.0097	0.0099	0.0097
skLinearSVC	[-3, -2, -1, 0]	22.4	0.09	0.8504	0.7404	0.7385	0.7394
				0.0061	0.0108	0.0109	0.0108
skLinearSVC	[0, 1]	15.5	0.08	0.8722	0.7781	0.7828	0.7804
				0.0039	0.0064	0.0063	0.0063
skLinearSVC	[0, 1, 2]	19.0	0.12	0.8693	0.7745	0.7766	0.7755
				0.0037	0.0065	0.0063	0.0063
skLinearSVC	[0, 1, 2, 3]	23.3	0.13	0.8635	0.7647	0.7657	0.7652
				0.0040	0.0070	0.0065	0.0068

Table 7.18: Results of the different Part of Speech-Taggers on the Language Slovak for the Universal Tagset

Tagger	Configuration	Time (seconds) Training	Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	141.6	13.01	0.7525 0.0076	0.5680 0.0131	0.6073 0.0139	0.5870 0.0135
nltkBrill	brill24	116.1	12.10	0.7516 0.0078	0.5673 0.0137	0.6069 0.0144	0.5864 0.0140
nltkBrill	nltkdemo18	93.4	11.89	0.7505 0.0077	0.5670 0.0135	0.6063 0.0143	0.5860 0.0139
nltkBrill	nltkdemo18plus	103.6	12.26	0.7506 0.0077	0.5671 0.0135	0.6064 0.0143	0.5861 0.0139
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	23.0	1.85	0.4523 0.0042	0.2171 0.0028	0.1926 0.0026	0.2041 0.0026
skMultinomialNB	[-2, -1, 0, 1, 2]	16.2	1.34	0.4899 0.0063	0.2488 0.0051	0.2229 0.0047	0.2352 0.0048
skMultinomialNB	[-1, 0, 1]	10.8	0.90	0.5293 0.0057	0.2775 0.0058	0.2559 0.0052	0.2662 0.0054
skMultinomialNB	[0]	4.9	0.52	0.5233 0.0075	0.2137 0.0065	0.2363 0.0068	0.2244 0.0067
skMultinomialNB	[-1, 0]	7.9	0.66	0.5397 0.0065	0.2827 0.0087	0.2706 0.0072	0.2765 0.0079
skMultinomialNB	[-2, -1, 0]	11.1	0.84	0.5170 0.0060	0.2688 0.0085	0.2488 0.0073	0.2584 0.0079
skMultinomialNB	[-3, -2, -1, 0]	13.5	1.10	0.4922 0.0050	0.2488 0.0059	0.2269 0.0048	0.2374 0.0053
skMultinomialNB	[0, 1]	8.1	0.72	0.5072 0.0049	0.2272 0.0041	0.2188 0.0040	0.2229 0.0040
skMultinomialNB	[0, 1, 2]	10.8	0.99	0.4781 0.0056	0.2115 0.0049	0.1956 0.0042	0.2033 0.0045
skMultinomialNB	[0, 1, 2, 3]	13.9	1.26	0.4551 0.0052	0.1980 0.0043	0.1802 0.0036	0.1887 0.0039
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	1512.5	1.59	0.6681 0.0074	0.4628 0.0086	0.4647 0.0101	0.4638 0.0093
skPerceptron	[-2, -1, 0, 1, 2]	1293.6	1.18	0.6761 0.0078	0.4817 0.0121	0.4860 0.0127	0.4838 0.0123
skPerceptron	[-1, 0, 1]	1078.6	0.76	0.6738 0.0083	0.4956 0.0102	0.4990 0.0108	0.4973 0.0104
skPerceptron	[0]	650.6	0.37	0.6729 0.0169	0.5129 0.0109	0.5673 0.0128	0.5387 0.0117
skPerceptron	[-1, 0]	888.5	0.59	0.6756 0.0110	0.5107 0.0090	0.5244 0.0098	0.5174 0.0094
skPerceptron	[-2, -1, 0]	1069.7	0.75	0.6816 0.0087	0.5095 0.0097	0.5203 0.0108	0.5148 0.0102
skPerceptron	[-3, -2, -1, 0]	1192.8	0.98	0.6842 0.0079	0.5038 0.0111	0.5131 0.0124	0.5084 0.0117
skPerceptron	[0, 1]	887.4	0.56	0.6526 0.0126	0.5091 0.0124	0.5128 0.0131	0.5109 0.0127
skPerceptron	[0, 1, 2]	1074.7	0.80	0.6523 0.0064	0.4898 0.0116	0.4795 0.0116	0.4846 0.0115
skPerceptron	[0, 1, 2, 3]	1170.1	1.01	0.6514 0.0065	0.4774 0.0099	0.4631 0.0094	0.4701 0.0095
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	386.1	0.67	0.7045 0.0077	0.5142 0.0104	0.5291 0.0121	0.5215 0.0112
skLinearSVC	[-2, -1, 0, 1, 2]	371.7	0.52	0.7196 0.0080	0.5444 0.0109	0.5631 0.0129	0.5536 0.0118
skLinearSVC	[-1, 0, 1]	493.3	0.34	0.7395 0.0073	0.5738 0.0122	0.6007 0.0135	0.5869 0.0128
skLinearSVC	[0]	1762.8	0.22	0.7250 0.0067	0.5413 0.0114	0.5987 0.0136	0.5686 0.0124
skLinearSVC	[-1, 0]	1028.0	0.26	0.7343 0.0065	0.5716 0.0117	0.6055 0.0129	0.5881 0.0122
skLinearSVC	[-2, -1, 0]	571.4	0.28	0.7308 0.0068	0.5671 0.0113	0.5966 0.0122	0.5815 0.0117
skLinearSVC	[-3, -2, -1, 0]	480.6	0.38	0.7237 0.0071	0.5546 0.0099	0.5813 0.0118	0.5676 0.0108
skLinearSVC	[0, 1]	986.3	0.24	0.7317 0.0081	0.5637 0.0125	0.6010 0.0136	0.5818 0.0130
skLinearSVC	[0, 1, 2]	573.5	0.34	0.7227 0.0073	0.5612 0.0122	0.5934 0.0131	0.5769 0.0126
skLinearSVC	[0, 1, 2, 3]	485.8	0.37	0.7176 0.0076	0.5487 0.0123	0.5791 0.0134	0.5635 0.0128

Table 7.19: Results of the different Part of Speech-Taggers on the Language Serbian for the MSD Tagset

Tagger	Configuration	Time (seconds) Training	Time (seconds) Tagging	Accuracy	Recall	Precision	F1-Score
				(second line: standard deviation)			
nltkBrill	fntbl37	80.7	10.33	0.8397 0.0059	0.6707 0.0107	0.6796 0.0115	0.6751 0.0111
nltkBrill	brill24	68.4	9.84	0.8383 0.0061	0.6686 0.0107	0.6785 0.0115	0.6735 0.0111
nltkBrill	nltkdemo18	45.7	9.68	0.8353 0.0062	0.6661 0.0111	0.6769 0.0119	0.6715 0.0115
nltkBrill	nltkdemo18plus	61.6	9.94	0.8373 0.0060	0.6679 0.0112	0.6780 0.0118	0.6729 0.0115
skMultinomialNB	[-3, -2, -1, 0, 1, 2, 3]	6.4	0.28	0.8356 0.0029	0.7398 0.0055	0.7019 0.0051	0.7204 0.0052
skMultinomialNB	[-2, -1, 0, 1, 2]	4.9	0.22	0.8510 0.0026	0.7549 0.0046	0.7267 0.0051	0.7405 0.0047
skMultinomialNB	[-1, 0, 1]	3.0	0.18	0.8651 0.0028	0.7647 0.0051	0.7506 0.0056	0.7576 0.0052
skMultinomialNB	[0]	1.2	0.09	0.8683 0.0047	0.7456 0.0088	0.7625 0.0094	0.7540 0.0090
skMultinomialNB	[-1, 0]	2.1	0.17	0.8617 0.0037	0.7476 0.0065	0.7432 0.0067	0.7454 0.0065
skMultinomialNB	[-2, -1, 0]	3.1	0.13	0.8508 0.0041	0.7391 0.0071	0.7244 0.0076	0.7316 0.0073
skMultinomialNB	[-3, -2, -1, 0]	3.8	0.18	0.8409 0.0043	0.7289 0.0068	0.7084 0.0075	0.7185 0.0071
skMultinomialNB	[0, 1]	2.1	0.10	0.8558 0.0024	0.7371 0.0053	0.7322 0.0055	0.7346 0.0052
skMultinomialNB	[0, 1, 2]	3.0	0.16	0.8461 0.0018	0.7278 0.0045	0.7140 0.0051	0.7209 0.0046
skMultinomialNB	[0, 1, 2, 3]	3.8	0.27	0.8360 0.0023	0.7150 0.0068	0.6953 0.0057	0.7050 0.0061
skPerceptron	[-3, -2, -1, 0, 1, 2, 3]	19.8	0.18	0.8358 0.0062	0.7104 0.0099	0.6904 0.0109	0.7003 0.0103
skPerceptron	[-2, -1, 0, 1, 2]	16.8	0.13	0.8415 0.0063	0.7264 0.0125	0.7082 0.0125	0.7172 0.0124
skPerceptron	[-1, 0, 1]	13.9	0.14	0.8393 0.0072	0.7407 0.0103	0.7288 0.0105	0.7347 0.0103
skPerceptron	[0]	7.8	0.06	0.8129 0.0137	0.6594 0.0107	0.6743 0.0113	0.6667 0.0109
skPerceptron	[-1, 0]	10.9	0.11	0.8097 0.0093	0.6745 0.0192	0.6657 0.0201	0.6701 0.0196
skPerceptron	[-2, -1, 0]	13.1	0.08	0.7983 0.0120	0.6657 0.0098	0.6477 0.0109	0.6566 0.0103
skPerceptron	[-3, -2, -1, 0]	15.3	0.09	0.7978 0.0116	0.6556 0.0142	0.6366 0.0145	0.6459 0.0143
skPerceptron	[0, 1]	11.2	0.09	0.8524 0.0085	0.7702 0.0113	0.7668 0.0112	0.7685 0.0112
skPerceptron	[0, 1, 2]	13.6	0.10	0.8448 0.0076	0.7407 0.0166	0.7307 0.0161	0.7356 0.0163
skPerceptron	[0, 1, 2, 3]	15.7	0.14	0.8411 0.0073	0.7247 0.0158	0.7117 0.0152	0.7182 0.0155
skLinearSVC	[-3, -2, -1, 0, 1, 2, 3]	28.1	0.22	0.8551 0.0039	0.7620 0.0078	0.7531 0.0079	0.7575 0.0077
skLinearSVC	[-2, -1, 0, 1, 2]	23.7	0.22	0.8703 0.0039	0.7897 0.0083	0.7880 0.0088	0.7888 0.0084
skLinearSVC	[-1, 0, 1]	17.1	0.09	0.8817 0.0034	0.8150 0.0058	0.8185 0.0058	0.8167 0.0058
skLinearSVC	[0]	24.4	0.06	0.8776 0.0034	0.7636 0.0057	0.7809 0.0063	0.7722 0.0060
skLinearSVC	[-1, 0]	17.0	0.06	0.8857 0.0041	0.7855 0.0081	0.7950 0.0086	0.7902 0.0083
skLinearSVC	[-2, -1, 0]	20.5	0.12	0.8830 0.0043	0.7841 0.0074	0.7897 0.0084	0.7869 0.0078
skLinearSVC	[-3, -2, -1, 0]	23.2	0.11	0.8755 0.0049	0.7716 0.0084	0.7739 0.0089	0.7727 0.0086
skLinearSVC	[0, 1]	17.8	0.10	0.8785 0.0078	0.7861 0.0067	0.7947 0.0071	0.7904 0.0069
skLinearSVC	[0, 1, 2]	19.7	0.12	0.8764 0.0040	0.7840 0.0060	0.7896 0.0065	0.7868 0.0062
skLinearSVC	[0, 1, 2, 3]	23.2	0.19	0.8732 0.0040	0.7762 0.0071	0.7796 0.0080	0.7779 0.0075

Table 7.20: Results of the different Part of Speech-Taggers on the Language Serbian for the Universal Tagset

References

[ABZ15] Thomas Stieglmaier Alexander Boehm and Thomas Ziegler. *Proposal: Integration and Evaluation of the MULTEX-Corpus in NLTK*, 2015.

[Ber] Robert C. Berwick. *An Idiot's Guide to Support Vector Machines.* Online available at `http://web.mit.edu/6.034/wwwbob/svm-notes-long-08.pdf`. [Online; Stand 6. September 2014].

[Bir06] Steven Bird. *NLTK: The Natural Language Toolkit.* Proceedings of the COLING/ACL on Interactive Presentation Sessions, COLING-ACL '06, pages 69–72, Stroudsburg, PA, USA, 2006. Association for Computational Linguistics.

[BKL09] Steven Bird, Ewan Klein, and Edward Loper. *Natural Language Processing with Python.* O'Reilly Media, 2009.

[Bri92] Eric Brill. *A Simple Rule-based Part of Speech Tagger.* Proceedings of the Workshop on Speech and Natural Language, HLT '91, pages 112–116, Stroudsburg, PA, USA, 1992. Association for Computational Linguistics.

[Bri95] Eric Brill. *Transformation-based Error-driven Learning and Natural Language Processing: A Case Study in Part-of-speech Tagging.* Comput. Linguist., volume 21, number 4, pages 543–565. MIT Press, December 1995.

[CL11] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: A library for support vector machines.* ACM Transactions on Intelligent Systems and Technology, volume 2, pages 27:1–27:27. 2011. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[Col02] Michael Collins. *Discriminative Training Methods for Hidden Markov Models: Theory and Experiments with Perceptron Algorithms.* Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10, EMNLP '02, pages 1–8, Stroudsburg, PA, USA, 2002. Association for Computational Linguistics.

[Con] Text Encoding Initiative Consortium. *TEI P5 Guidelines*. Online available at `http://www.tei-c.org/index.xml`. [Online; Stand 6. September 2014].

[EBD+10] Tomaž Erjavec, Ana-Maria Barbu, Ivan Derzhanski, Ludmila Dimitrova, Radovan Garabík, Nancy Ide, Heiki-Jaan Kaalep, Natalia Kotsyba, Cvetana Krstev, Csaba Oravecz, Vladimír Petkevič, Greg Priest-Dorman, Behrang QasemiZadeh, Adam Radziszewski, Kiril Simov, Dan Tufiş, and Katerina Zdravkova. *MULTEXT-East "1984" annotated corpus 4.0*, 2010. Online available at `http://hdl.handle.net/11356/1043`. Slovenian language resource repository CLARIN.SI.

[Erj] Tomaž Erjavec. *MULTEXT-East Home Page*. Online available at `http://nl.ijs.si/ME/`. [Online; Stand 6. September 2014].

[Kri07] David Kriesel. *A Brief Introduction to Neural Networks*. 2007.

[Lin] Chih-Jen Lin. *Support Vector Machines - Talk at Machine Learning Summer School*. Online available at `http://www.csie.ntu.edu.tw/~cjlin/talks/MLSS.pdf`. [Online; Stand 6. September 2014].

[Mil95] George A. Miller. *WordNet: A Lexical Database for English*. Commun. ACM, volume 38, number 11, pages 39–41. ACM, November 1995.

[MN98] Andrew McCallum and Kamal Nigam. *A comparison of event models for Naive Bayes text classification*. IN AAAI-98 WORKSHOP ON LEARNING FOR TEXT CATEGORIZATION, pages 41–48. AAAI Press, 1998.

[MRS08] Christopher D Manning, Prabhakar Raghavan, and Hinrich Schütze. *Introduction to information retrieval*. Cambridge University Press, 2008.

[PDM11] Slav Petrov, Dipanjan Das, and Ryan T. McDonald. *A Universal Part-of-Speech Tagset*. CoRR, volume abs/1104.2086, . 2011.

[PVG+11] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. *Scikit-learn: Machine Learning in Python*. Journal of Machine Learning Research, volume 12, pages 2825–2830. 2011.

[Roj96] Rau Rojas. *Neural networks: a systematic introduction*. Springer, 1996.

[Seb02] Fabrizio Sebastiani. *Machine Learning in Automated Text Categorization*. ACM Comput. Surv., volume 34, number 1, pages 1–47. ACM, March 2002.

[Spe04] C. Spearman. *The Proof and Measurement of Association between Two Things*. The American Journal of Psychology, volume 15, number 1, pages pp. 72–101. University of Illinois Press, 1904.

[Vou03] Atro Voutilainen. *Part-of-speech tagging*. Oxford University Press, 2003.

[ZW05] Fei Zheng and Geoffrey I. Webb. *A comparative study of semi-naive Bayes methods in classification learning*. Proc. 4th Australasian Data Mining conference (AusDM05, pages 141–156, 2005.