Automated Learning of Hungarian Morphology for Inflection Generation and Morphological Analysis

Gábor Szabó[1], László Kovács[2]
[1,2]Institute of Information Technology, University of Miskolc, Hungary

Article Info

Article historys:

Received Jun 15, 2020
Revised Nov 22, 2020
Accepted Dec 2, 2020

Keywords:

Morphology
Agglutination
Inflection generation
Morphological Analysis
Automated Learning

ABSTRACT

The automated learning of morphological features of highly agglutinative languages is an important research area for both machine learning and computational linguistics. In this paper we present a novel morphology model that can solve the inflection generation and morphological analysis problems, managing all the affix types of the target language. The proposed model can be taught using *(word, lemma, morphosyntactic tags)* triples. From this training data, it can deduce word pairs for each affix type of the target language, and learn the transformation rules of these affix types using our previously published, lower-level morphology model called ASTRA. Since ASTRA can only handle a single affix type, a separate model instance is built for every affix type of the target language. Besides learning the transformation rules of all the necessary affix types, the proposed model also calculates the conditional probabilities of the affix type chains using relative frequencies, and stores the valid lemmas and their parts of speech. With these pieces of information, it can generate the inflected form of input lemmas based on a set of affix types, and analyze input inflected word forms. For evaluation, we use Hungarian data sets and compare the accuracy of the proposed model with that of state of the art morphology models published by SIGMORPHON, including the Helsinki (2016), UF and UTNII (2017), Hamburg, IITBHU and MSU (2018) models. The test results show that using a training data set consisting of up to 100 thousand random training items, our proposed model outperforms all the other examined models, reaching an accuracy of 98% in case of random input words that were not part of the training data. Using the high-resource data sets for the Hungarian language published by SIGMORPHON, the proposed model achieves an accuracy of about 95-98%.

Corresponding Author:

Gábor Szabó,
Institute of Information Technology,
University of Miskolc,
Miskolc-Egyetemváros, H 3515, Hungary,
Email: szgabsz91@gmail.com

1. INTRODUCTION

According to the theory of morphology and computational linguistics, words are built up from morphemes, that are the smallest morphological units with associated meaning [1]. The grammatically correct root form of a word is called the lemma, while the added morphemes that modify its base meaning are called affixes. In morphologically complex languages, affixes may change some of the characters in the root form as well, resulting in for example vowel or consonant gradation.

The process of adding affixes to a word is called inflection, while the inverse operation when we determine the lemma and the affixes of a word is called morphological analysis.

In natural languages there are a finite number of affix types that determine the semantic meaning of the affixes, i.e. how the meaning of the base form is altered by them. Examples of affix types include accusative case, plural form, past tense, etc. The concrete appearance of affix types are affixes in the words.

Another important morphological feature of a word is its part of speech (POS) that indicates its main syntactic feature. One word might have multiple possible parts of speech, and the part of speech of a word might change during inflection, when using derivative affix types. As an example, the word *"good"* is an adjective, but its inflected form *"goodness"* is a noun.

Natural languages represent different levels of morphological complexity. The target language of our research is Hungarian, a morphologically complex agglutinative language. In Hungarian, there are a large number of affix types and each word can contain many affixes that often alter the base form of the word.

The goal of our research is to create a novel morphology model that can learn the morphology of highly agglutinative languages in an automated way, and then generate inflected word forms from a lemma and a set of affix types, as well as morphologically analyze input inflected words, providing not only its lemma, part of speech and affix types, but also its intermediate inflected word forms.

Morphological analysis is a very complex problem, because we cannot know which affix types to search for in the provided inflected word form, or even how many affix types it contains. That's why many of the models found in literature solves a simpler problem called morphological segmentation. During segmentation, the goal is to find the affix boundaries in the input word, but usually the affix types themselves are not identified. The most widely used segmentation models are part of the Morfessor model family published by Creutz and Lagus [2] and Virpioja et al. [3]. However, segmentation models cannot reach high accuracy with agglutinative languages, because the base word forms are often altered by the added affixes, blurring the affix boundaries.

Lately, SIGMORPHON (Special Interest Group on Computational Morphology and Phonology) publishes different shared tasks in the area of morphology and phonology that can be solved using novel morphology models. The best models are published in SIGMORPHON's annual proceedings. One of the common shared tasks has been inflection generation, for which the training and test data sets are provided by SIGMORPHON. Among the published models we collected those ones whose source code could be found online, and we used these models during the evaluation of our proposed model.

From the SIGMORPHON 2016 publication [4], the Helsinki model [5] was selected. It uses a one dimensional residual network architecture with constant size across the layers, followed by either one or zero Gated Recurrent Unit layers. The output vector of each residual layer is combined with the vector of the previous layer by addition, which means that the output is the sum of the input and the output of each layer.

From the SIGMORPHON 2017 publication [6], we selected two models. The UF method [7] models the morphological reinflection problem using an encoder-decoder architecture. For an input word, every character is encoded through a Bi-directional Gated Recurrent Unit (GRU) network. Another GRU network is deployed as a decoder to generate the inflection. The UTNII model [8] is also based on the seq2seq model, and with its configuration, it was the second best model in 2017 using the high-resource data sets. Since 2017, all winner models contained some kind of artificial intelligence parts. Also, new data sets were published for training that contained less and less information about the target language. The trend is that most models achieved good results using the high-resource data sets, but their accuracy dropped significantly for medium-resource and low-resource data sets.

2018 was the last year when the morphological inflection problem was among the shared tasks [9]. After that, sentential context and cross-language problems became the main focus. Among the published models in 2018, we selected three entries. The Hamburg model [10] introduces the concept of patches that act as string transducer actions. The resulting model is a language-agnostic network model that aims to reduce the number of learned edit operations by introducing equivalence classes over graphical features of individual characters. The IITBHU model [11] uses a Pointer-Generator Network to mitigate the problem of copying many characters between word forms. The lemma and the morphosyntactic tags are encoded by two separate encoders. Compared to other similar performing systems, this model is trained end-to-end, doesn't require data augmentation techniques, and uses soft attention over hard monotonic attention, making the resulting system more flexible. The MSU model [12] aimed to improve the accuracy in medium and low-resource scenarios by explicitly equipping the decoder with the information from the character-based language model, however the advantage was not clear.

These six models will be used as baseline models during evaluation. Our main evaluation will be based on our own data sets that have been collected from the Internet, but for cross-validation purposes, we will also evaluate our proposed model using the data sets provided by SIGMORPHON. Our goal is to design a novel morphology model that can match or even outperform the accuracy and generalization capability of the above mentioned base models. In the current state of our research, we won't include any artificial intelligence techniques, to see how well our proposed model can perform using classical pattern matching methods against the six SIGMORPHON models that are all based on AI components.

2. RESEARCH METHOD

This paper's proposed morphology model is an extension of our previously published, single-affix morphology model called ASTRA (Atomic String Transformation Rule Assembler) [13].

The goal of ASTRA is to learn the transformation rules of a single affix type from a provided set of training word pairs. These rules have a very simple structure, they only contain a prefix and postfix component, as well as the changing substring and a replacement string. With this kind of rule structure, ASTRA can transform a provided base word form to its inflected form, and it can also return the base form of an inflected word.

For example if we train an ASTRA model instance using a word pair set of English word pairs demonstrating the transformation rules of the plural affix type, then the resulting ASTRA instance will be able to return *"apples"* if the input word is *"apple"*, or if we want to produce the base form of *"tables"*, it will correctly yield the word *"table"*. The details of the ASTRA model including its rule model, training algorithm, inflection method and evaluation can be found in [13].

According to the evaluation, ASTRA reached exceptional accuracy (more than 94%) using word pairs for the Hungarian accusative case. Its average training and search times were also very low, especially when we incorporated prefix trees for the storage of the generated rules. Therefore, ASTRA is a perfect candidate to use for the learning of transformation rules for a single affix type.

In this paper we build a new morphology model on top of ASTRA. The base concept behind this extension is that since ASTRA can only handle a single affix type, let's have a separate ASTRA model instances for each affix type of the target language. This way we can manage all the affix types of the language, and learn their transformation rules separately. The proposed model also identifies the possible affix type chains and calculates their conditional probability so that it can decide which affix type can be added after which other affix types. The valid lemmas of the language are also stored, as well as their possible parts of speech.

The complexity of the proposed model comes from the fact that in agglutinative languages, there are a large number of affix types, and the number of possible affix type chains can also vary, as well as the length of these chains. Figure 1 displays the affix type chains found in our generated data sets, where every dot is an occurrence of an affix type, and the lines define the adjacency among these affix type occurrences. (The process of the data generation will be described later in Subection 3.1 in more details.) There are 35,954 different affix type chains in this data set (represented by 35,954 line paths in the figure), the longest ones contain 9 affix types (meaning that the longest paths contain 9 dots), while the median of the affix type chain lengths is 5. The graph in Figure 1 looks rather abstract, since there are so many variations that our eyes cannot distinguish the different paths properly.

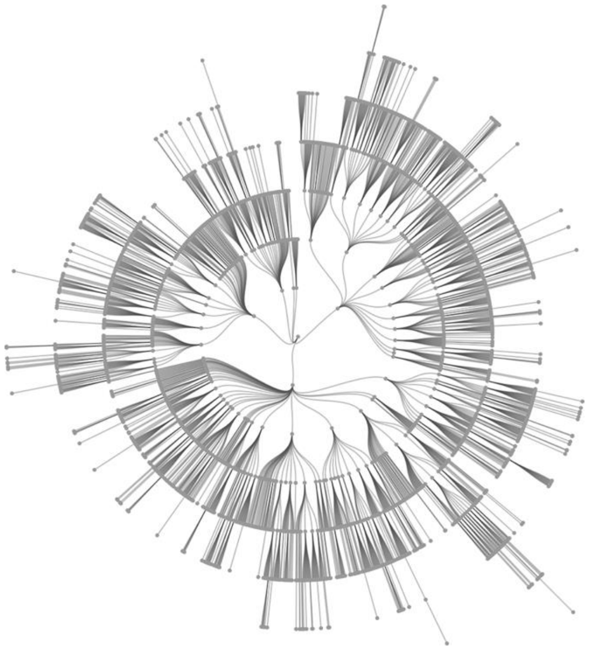

Figure 1. The visual representation of all the affix type chains in the Hungarian language

As the above examples demonstrate, we can define an adjacency relation among the different affix types. If $t_i, t_j \in T$ are affix types, then $t_i \rightarrow t_j$ denotes that the affix type t_j can be applied on a word whose last affix type is t_i, i.e. t_j can come after t_i. Not every affix type can come after any other affix types. For example, in the Hungarian language, past tense can only be applied to verbs and not nouns; and accusative case can come after plural, but not vice versa.

In this sense, the vocabulary of a language can be thought of as a word graph. The base of this graph are the lemmas that don't contain any affix types, and we get an inflected word form if we add affix types to these lemmas. According to this approach, it can be easily seen that every word is reachable starting from a lemma, applying a number of affix types. Let's assume indirectly that there is at least one word for which this proposition is not true. If this word contains k affix types and we start removing these affix types one by one, we get a reversed word form chain, where the last word form has no affix types. Since it has no affixes, it is a lemma, and during the affix removal process, we got a word form chain, whose starting point is a lemma. This is a contradiction, meaning that the original proposition was true.

This word graph can reach a huge size for natural languages that makes it more complex to learn the morphology of agglutinative languages. Another problem that needs to be adressed by morphology models is that one word form might have different possible morphological structures, and based on a single affix type, it might have different inflected word forms. Figure 2 displays a simple example for this case: the Hungarian word *"oszlat"* is a causative inflected word form that has two possible root forms: *"oszol"* and *"oszlik"* that both mean *"decay"*. If we add the subjunctive imperative affix type to *"oszlat"*, we can get two different word forms as well: *"oszlasson"* or *"oszlassék"*, the latter one being an archaic word form.

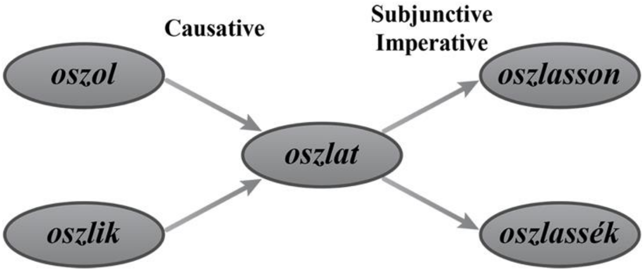

Figure 2. A Hungarian word having multiple lemmas and inflected word forms

This means that any morphology model that aims to reach high accuracy in case of the Hungarian language must be able to traverse the above affix type graph efficiently.

2.1. Training Phase

The training data of the proposed model is a set of *(word, lemma, morphosyntactic tags)* triples. Let's denote words by $w \in W$, while lemmas by $\overline{w} \in \overline{W} \subset W$. The list of morphosyntactic tags contains both the part of speech, and all the affix types found in the word. Affix types will be denoted by $t \in T$, while parts of speech will be denoted by $\overline{t} \in \overline{T}$.

The first problem to be solved during the training phase is to deduce training word pairs for the ASTRA model instances that demonstrate the transformation rules of each affix type. For the ASTRA instance that must learn the rules of the affix type t, the word pairs must be in the form of (w_1, w_2) where w_2 contains the same affix types as w_1, plus an additional affix type (t).

In case of training records where the word contains only one affix type, the word pair will consist of the lemma and the inflected word form. In other cases, if the inflected form of the training record contains the affix types $t_1, \dots, t_k, t$, then we must also find another training record with the same lemma that only contains the affix types $t_1, \dots, t_k$. From these two records, we can deduce a word pair from the two inflected word forms for the affix type t.

Table 1. Sample training records demonstrating the word pair generation process

Index	Inflected word	Lemma	Morphosyntactic tags
1	almák	alma	Noun, Plural
2	labdák	labda	Noun, Plural
3	almákat	alma	Noun, Plural, Accusative case
4	labdát	labda	Noun, Accusative case
5	almát	alma	Noun, Accusative case

As an example, let’s take a look at the training records of Table 1. Here, a couple of inflected word forms are presented related to the Hungarian words *“alma”* (*“apple”* in English) and *“labda”* (*“ball”* in English).

These records are processed after grouping them by their lemmas. From the records with index 1, 2, 4 and 5, we can directly deduce the word pairs of *(alma, almák)* and *(labda, labdák)* for plural, as well as *(labda, labdát)* and *(alma, almát)* for accusative case, since they only contain one affix type.

The training record with index 3 contains two affix types (plural and accusative case), so we need to find another record that has the same lemma (*alma*) and contains only the plural affix type. This other training record is the first one, whose inflected form will be the first component of the deduced word pair: *(almák, almákat)*.

This way we can generate a training word pair set for each affix type in the original training data, and have a separate ASTRA model instance learn the transformation rules of these affix types as demonstrated in [13].

Another task that needs to be solved during the training of the proposed model is to calculate the conditional probabilities of the found affix type chains. This is achieved using relative frequencies. The M function can return the conditional probability of any affix type chains starting from a part of speech in the following way:

$$M(\bar{t}_0, t_1, \dots, t_i) = \begin{cases} P(\bar{t}_0)\ if\ i = 0 \\ P(\bar{t}_0) \cdot \prod_{j=1}^{i} P\big(t_j \mid \bar{t}_0, t_1, \dots, t_{j-1}\big)\ otherwise \end{cases} \tag{1}$$

Here, the conditional probability of a single step can be calculated in the following way:

$$P(t_n \mid \bar{t}_0, t_1, \dots, t_{n-1}) = \frac{P(\bar{t}_0 \cap t_1 \cap \dots \cap t_{n-1} \cap t_n)}{P(\bar{t}_0 \cap t_1 \cap \dots \cap t_{n-1})} \tag{2}$$

This means that we must divide the number of words that contain all the $n + 1$ morphosyntactic tags by the number of words that do not contain t_n.

If M outputs 0 for a given affix type chain, it means that at least one affix type in the chain cannot come after its predecessors.

2.2. Inflection Generation

The proposed model uses separate ASTRA model instances for each affix type in the target language. This means that during inflection generation, these ASTRA instances will produce the inflected word forms.

The input of inflection generation is a lemma and a set of affix types. The goal is to produce the inflected word form whose lemma is the provided input lemma, and that has all the provided affix types in a valid order. The model’s responsibility is to determine all the valid orders of these affix types and have the appropriate ASTRA instances transform the current word form, one by one.

Let’s denote the transformation operator of the ASTRA instance related to the affix type t by C_{forw}^{t}. For example if the affix type t is plural and the input word is apple, then $C_{forw}^{t}(apple) = apples$. Using this operator, we can define the inflection generation operator of the proposed model as

$$Infl: \bar{W} \times \{t_1, \dots, t_m\} \rightarrow \{(\bar{t}_0, \langle S_1, \dots, S_m \rangle, c_i)\} \tag{3}$$

where $\langle S_i \rangle$ denotes a list of items with fixed order and c_i is a confidence value. The list containing S_i items contain the individual steps of the inflection generation:

$$S_i = \left(t_i,\ C_{forw}^{t_i}(w_{i-1})\right) \tag{4}$$

Summarizing the above formulae, the inflection generation operator of the proposed model has the following features:

- The input of the operator is a lemma $\bar{w}_0$ and a set of m affix types.
- The first task is to determine the valid order of the provided affix types using the previously calculated conditional probability values. This step results in a permutation of the provided affix types. This affix type chain must be valid for at least one of the possible POS of the provided lemma ($\bar{t}_0$).

- Then, the appropriate ASTRA model instance is used to produce the intermediate word forms: the ASTRA instance related to the affix type t_i will transform the previous step's word form w_{i-1}. This way we produce the $w_1, \dots, w_{m-1}$ intermediate word forms.
- The final step will produce the w_m final inflected word form that contains all the provided affix types in a valid order.

The confidence value denotes how strong an answer is. If there are multiple responses, they are returned in reversed confidence order, meaning that the response with the highest confidence will be the first.

As an example, if the input lemma is *"alma"* and the set of affix types contains accusative case and plural, then the proposed model first determines that the valid order of the affix types is (plural, accusative case). It also knows from the lemma store that the given lemma is a noun. The input word form is then transformed first using the ASTRA instance related to plural, then the ASTRA instance related to accusative case, producing the word form chain *(plural, almák), (accusative case, almákat)*.

2.3. Morphological Analysis

For morphological analysis, a similar operator can be constructed. Let's denote the backwards transformation operator of the ASTRA model instance related to the affix type t by C_{backw}^{t}. This operator can produce a word form from an inflected word whose last affix type is t. The output will not have this t affix type, so this operator can be used during morphological analysis. For instance, if t is plural, then $C_{backw}^{t}(apples) = apple$.

The input of morphological analysis is an inflected word form. The goal is to determine its lemma and identify all the affix types found in the input word. Morphological analysis represents a more complex problem than inflection generation, since we don't know which affix types to search for, or even how many affixes the input word contains. This means that the proposed model needs to check much more possibilities during morphological analysis than during inflection generation. The model tries to remove the last affix type of the current word form using the ASTRA model instances. If the resulting word form is present in the lemma store of the proposed model, then the morphological analysis process can be stopped, as the model found the lemma.

Using ASTRA's C_{backw}^{t} operator, we can define the morphological analysis operator of the proposed model as

$$Ana: W \rightarrow \{(\langle S_m, \dots, S_1 \rangle, \bar{t}_0, c_i)\} \tag{5}$$

Where $\langle S_i \rangle$ denotes a list of items with fixed order and c_i is a confidence value. The list containing S_i items contain the individual steps of the morphological analysis:

$$S_i = \left(t_i, C_{backw}^{t_i}(w_i)\right) \tag{6}$$

Summarizing the above formulae, the morphological analysis operator of the proposed model has the following features:

- The input of the operator is an inflected word form w_m.
- The model tries to identify the affix types of the input word, and it always removes the last one, one after the other using the appropriate ASTRA instances. This will result in an intermediate word form chain $w_{m-1}, \dots, w_1$.
- The word form w_1 will only contain the affix type t_1. This word form is transformed using the ASTRA instance related to t_1, producing the lemma $\bar{w}_0$. The appropriate POS is retrieved from the lemma store of the model.
- If the model finds the lemma, it knows it can stop.

During morphological analysis, the conditional probabilities help the model to only choose affix types that make up a valid affix type chain.

The confidence value denotes how strong an answer is. If there are multiple responses, they are returned in reversed confidence order, meaning that the response with the highest confidence will be the first.

As an example, if the input word is *"almákat"*, then the proposed model tries to identify the last affix type found in the word. When it checks accusative case, the related ASTRA instance can produce the word form *"almák"* by removing the accusative case affix type. After that, the ASTRA instance related to plural will be able to produce a new word form *"alma"*. As this word form can be found in the lemma store, the proposed model knows that it can stop and the POS is noun. Of course, much more affix types are checked along the way, but they won't lead to a valid chain, being unable to produce a subsequent word form at one point. Only those chains will be returned that lead to a valid, known lemma.

3. RESULTS AND DISCUSSION

In this section we evaluate the proposed model, comparing it with existing baseline models, measuring different metrics.

In Subsection 3.1 we use custom, generated Hungarian data sets for training and evaluating the examined models. We measure their average accuracy and the size of their knowledge bases. We also examine the average inflection and analysis times of the proposed model.

In Subsection 3.2 we try to measure the generalization capabilities of the examined models using 100 artificial words. The theory behind this test is to check if the models can handle non-meaningful artificial words that imitate the transformation rules of real affix types. If they can handle these words after training them with real training data, it means that the models can generalize well.

In Subsection 3.3 we perform cross-validation, using the data sets of SIGMORPHON to see how well the proposed model inflects the words published by SIGMORPHON. This way we can compare our model directly with the published results of the SIGMORPHON models.

3.1. Comparison with Existing Models Using Generated Hungarian Data Sets

In this subsection we use data sets that we had generated previously using an automated process [14]. The algorithm that we developed for generating large volumes of training and evaluation data used the documents of the Hungarian Electric Library as the data source, in order to extract the words out of these documents.

From the 16,250 documents of this site, we could gather 13,345,903 unique word candidates. These items were processed using Hunmorph [15], which is a very efficient and accurate morphological analyzer for the Hungarian language [16]. Its language dependent rule database is called Morphdb.hu [17].

Hunmorph produced 4,423,882 different morphological structures, covering 2,515,570 unique word forms. For the evaluation in this subsection, we selected random items from this data set to train the examined models, then we selected random, disjoint evaluation data sets to evaluate them. The number of training items was 10 thousand, 20 thousand, …, 100 thousand; while the number of evaluation items was constantly 10 thousand. The training and evaluation data sets were always disjoint.

We compared the proposed model with available baseline models submitted as part of previous SIGMORPHON tasks: Helsinki (2016), UF and UTNII (2017), Hamburg, IITBHU and MSU (2018), all of which are AI based morphology models.

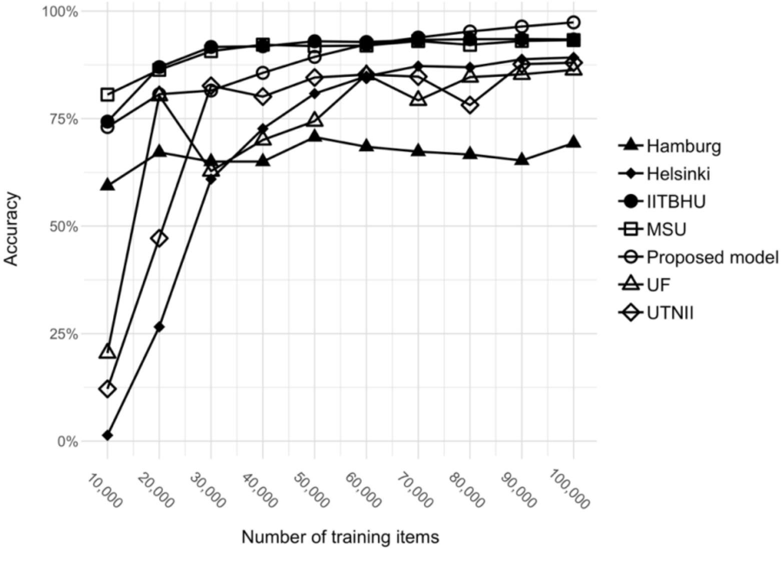

Figure 3. The accuracy of the examined models

The first metric that we examined was the average accuracy, i.e. how many evaluation words were inflected/analyzed correctly by the models. The results are summarized by Figure 3.

As we can see, our proposed novel model achieved the highest accuracy using 100 thousand training items, reaching about 98%. The Hamburg model performed the worst with about 70%. The other SIGMORPHON models achieved more than 85%. The MSU and IITBHU models performed especially well, with more than 93%.

We can also observe the steepness of these curves. Some of the SIGMORPHON models started from a very low percentage, including Helsinki, UTNII and UF. They achieved less than 25% using 10 thousand training items. In contrast, our proposed model starts from about 75%, which shows that the proposed model can learn the main morphological correlations even after being trained using a relatively small training data set. The proposed model overtakes the two best SIGMORPHON models (MSU, IITBHU) at around 60 thousand training items.

The downside of the proposed model is that it cannot run on a GPU, while the SIGMORPHON models can. If we examine the chart in Figure 4, we can see that its average analysis time is much higher than its average inflection time. This is due to the fact that while the input of the inflection generation operation contains the required affix types, during morphological analysis the model needs to check much more cases to find the affix types in the input word. Neither the set of affix types, nor the number of affix types is known.

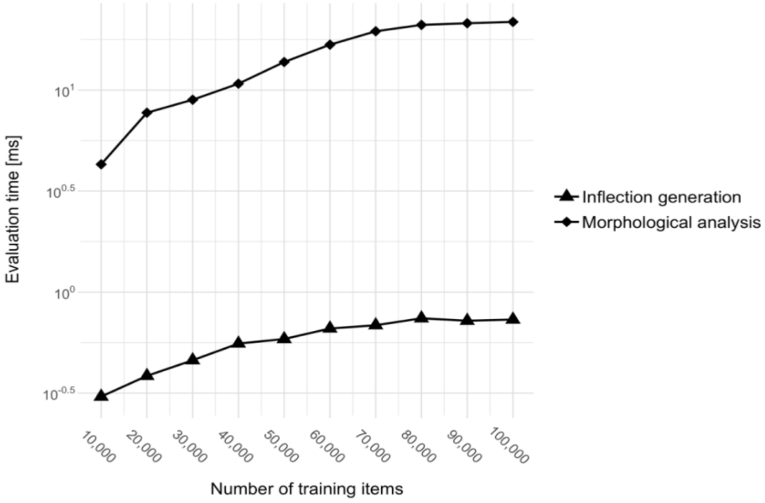

Figure 4. The average time of inflection generation and morphological analysis

We also measured the file size of the exported knowledge bases, these values can be seen in Table 2. As we can see, MSU and UF produce smaller files, and our proposed model is the third best. We also included Hunmorph that has a much larger knowledge database, about three times as big as our proposed model. The Helsinki and UTNII models also have very large exported knowledge bases.

Table 2. The file size of the exported knowledge bases

Model	File size
Proposed model	7.8 MB
Hunmorph	22.7 MB
Helsinki	58.3 MB
UF	4.5 MB
UTNII	92.4 MB
IITBHU	8.3 MB
MSU	1.5 MB

3.2. Verifying the Generalization Capabilities

As a next step, we wanted to compare the generalization capabilities of the examined models. To do this, we first generated random artificial words, then manually created their inflected word forms using the transformation rules of Hungarian accusative case. The random words were generated by an automated algorithm that combined 3-6 syllables randomly from real Hungarian words. The following list contains three of the resulting random word pairs:

- *abajkasztell – abajkasztellt*
- *medarkónunkgótpüf – medarkónunkgótpüföt*
- *öldberczerinc – öldberczerincet*

The theory behind this evaluation test is that if a model has a good generalization capability, then it will be able to handle these artificial words as if they were real, meaningful words. The models have been trained by the same training data set as in the previous subsection.

Table 3 contains the results of this test. As we can see, some of the SIGMORPHON models couldn’t handle the artificial words at all. Only the Helsinki model could achieve an accuracy of 41%. Hunmorph was the second best model with 89%, but our novel proposed model could handle the most words correctly, with an accuracy of 95%.

Table 3. The accuracy of the examined models using 100 artificial words

Model	Accuracy
Proposed model	95%
Hunmorph	89%
Helsinki	41%
UF	0%
UTNII	0%
Hamburg	0%
IITBHU	0%
MSU	0%

3.3. Cross-Validation Using the SIGMORPHON Data Sets

We also wanted to compare our novel morphology model with the SIGMORPHON models using the data sets provided by SIGMORPHON. In 2016, a single training data set was published for Hungarian, but in 2017 and 2018 there were three separate training data sets: a high-resource, a medium-resource and a low-resource training data set.

Most published models that could achieve exceptional accuracy using the high-resource data set lost many percent points in case of the medium-resource and low-resource data sets. Since our research goals don’t include any limitations on the size of the training data sets, we expected similar drops in the accuracy.

The results are contained by Table 4. As we can see, the proposed model reached 95-98% in case of the high-resource training data sets, that dropped to about 50-60% in case of the medium-resource and low-resource training data sets.

According to the SIGMORPHON publications, in 2017 the CLUZH and the LMU models were the best models that reached about 86% in case of Hungarian, and in 2018 the UZH model won with about 87%. These results make our proposed models exceptionally accurate in case of the Hungarian language. This is especially interesting, considering that we did not apply any artificial intelligence methods yet, unlike the six examined SIGMORPHON models.

Table 4. Accuracy of the proposed model using the SIGMORPHON data sets

Data set	Accuracy
SIGMORPHON 2016	98.03%
SIGMORPHON 2017 low	49.64%
SIGMORPHON 2017 medium	54.40%
SIGMORPHON 2017 high	95.14%
SIGMORPHON 2018 low	53.69%
SIGMORPHON 2018 medium	59.92%
SIGMORPHON 2018 high	95.43%

4. CONCLUSION

In this paper we presented a novel multi-affix morphology model that can learn the morphology of highly agglutinative languages like Hungarian, and solve the inflection generation and morphological analysis problems, managing all the affix types of the target language. The proposed model can be trained using *(word, lemma, morphosyntactic tags)* triples. During the training phase, the proposed model calculates the conditional probability of all the possible affix type chains, stores the valid lemmas and their parts of speech, and trains a separate ASTRA model instance for each affix type, using a deduced set of word pairs demonstrating the transformation rules of the target affix type. This way, the proposed model can later generate inflected word forms based on a given lemma and a provided set of affix types, and it can analyze input inflected word forms.

Our evaluation showed that the proposed model can achieve exceptionally high average accuracy (98%), even though it does not include any AI components yet. It was compared with six state of the art AI based SIGMORPHON models including Helsinki (2016), UF and UTNII (2017), as well as Hamburg, IITBHU and MSU (2018). As for the size of the knowledge base, the proposed model had the third smallest exported file. The high generalization capability of our novel model was justified using generated artificial words, among which the proposed model could handle 95% correctly, which was the highest value among the examined models. Using the data sets published by SIGMORPHON, the proposed model achieved an accuracy of 95-98% for high-resource scenarios, while its accuracy dropped to about 50-60% for the medium-resource and low-resource data sets.

The presented results are very promising, however, there are still much room for improvements. As we saw, the average analysis time is higher than the average inflection time. In order to reduce the difference between the two evaluation times, we could further optimize the proposed model.

One way to decrease the runtime of the analysis operation is to eliminate parts of the knowledge base, for instance those rules in the ASTRA instances that don't hold much information about the training data. Dropping such rules wouldn't degrade the average accuracy, but it would decrease the search time inside the ASTRA instances significantly.

Another optimization technique could be to introduce some kind of artificial intelligence methods, for example to train a neural network that could determine the last affix type of a given word form. Although this would increase the average training time of the model, with this knowledge, many of the cases could be eliminated during morphological analysis. This way the average analysis time would decrease radically, since the model could concentrate on those affix types that have higher probability based on the current word form.

REFERENCES

[1] L. Bauer, *Introducing Linguistic Morphology*, 2nd ed. Edinburgh: Edinburg University Press, 2003.

[2] M. Creutz *et al.*, "Inducing the morphological lexicon of a natural language from unannotated text," in *Proceedings of the International and Interdisciplinary Conference on Adaptive Knowledge Representation and Reasoning (AKRR'05)*, pp. 106-113, 2005.

[3] S. Virpioja, P. Smit, S. Grönroos and M. Kurimo, *Morfessor 2.0: Python implementation and extensions for Morfessor Baseline*", Aalto University publication series, 2013.

[4] R. Cotterell *et al.*, "The SIGMORPHON 2016 shared task – morphological reinflection," in *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pp. 10-22, 2016.

[5] R. Östling, "Morphological reinflection with convolutional neural networks," in *Proceedings of the 14th SIGMORPHON Workshop on Computational Research in Phonetics, Phonology, and Morphology*, pp. 23-26, 2016.

[6] R. Cotterell *et al.*, "CoNLL-SIGMORPHON 2017 shared task: Universal morphological reinflection in 52 languages," in *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pp. 1-30, 2017.

[7] Q. Zhu *et al.*, "Character sequence-to-sequence model with global attention for universal morphological reinflection," in *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pp. 85-89, 2017.

[8] H. Senuma *et al.*, "Seq2seq for Morphological Reinflection: When Deep Learning Fails," in *Proceedings of the CoNLL SIGMORPHON 2017 Shared Task: Universal Morphological Reinflection*, pp. 100-109, 2017.

[9] R. Cotterell *et al.*, "The CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection," in *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pp. 1-27, 2018.

[10] F. Schröder *et al.*, "Finding the way from ä to a: Sub-character morphological inflection for the SIGMORPHON 2018 Shared Task," in *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pp. 76-85, 2018.

[11] A. Sharma *et al.*, "IIT (BHU) – IIITH at CoNLL–SIGMORPHON 2018 Shared Task on Universal Morphological Reinflection," in *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pp. 105-111, 2018.

[12] A. Sorokin, "What can we gain from language models for morphological inflection?" in *Proceedings of the CoNLL–SIGMORPHON 2018 Shared Task: Universal Morphological Reinflection*, pp. 99-104, 2018.

[13] L. Kovács and G. Szabó, "String Transformation Based Morphology Learning," *Informatica*, vol. 43, no. 4, pp. 467-476, 2019.

[14] G. Szabó *et al.*, "Efficiency Analysis of Inflection Rule Induction," in *Proceedings of the 2015 16th International Carpathian Control Conference (ICCC)*, pp. 521-525, 2015.

[15] V. Trón *et al.*, "Hunmorph: open source word analysis," in *Proceedings of the Workshop on Software*, pp. 77-85, 2005.

[16] G. Szabó and L. Kovács, "Benchmarking morphological analyzers for the Hungarian language," *Annales Mathematicae et Informaticae*, vol. 49, pp. 141-166, 2018.

[17] V. Trón *et al.*, "Morphdb.hu: Hungarian lexical database and morphological grammar," in *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, 2006.

BIOGRAPHY OF AUTHORS

Gábor Szabó is currently a PhD student at the Institute of Information Technology at the University of Miskolc, Hungary. He is a BSc in Software Information Technology since 2012 and an MSc in Engineering Information Technology since 2014. His main research area is the automated learning of the morphological features of morphologically complex, highly agglutinative languages, in order to solve the inflection generation and morphological analysis problems.

László Kovács is a Professor of the Institute of Information Technology at the University of Miskolc, Hungary. He received his PhD in computer science in 1998. His broad research areas include morphology, ontologies and database systems among others.