Enhancing Turkish Word Segmentation: A Focus on Borrowed Words and Invalid Morpheme

Soheila Behrooznia
Dept. of CS and IT
IASBS
Zanjan, Iran
s.behrooznia@iasbs.ac.ir

Ebrahim Ansari
Dept. of CS and IT
IASBS
Zanjan, Iran
ansari@iasbs.ac.ir

Zdeněk Žabokrtský
ÚFAL
Charles University
Prague, Czechia
zabokrtsky@ufal.mff.cuni.cz

Abstract

This study addresses a challenge in morphological segmentation: accurately segmenting words in languages with rich morphology. Current probabilistic methods, such as Morfessor, often produce results that lack consistency with human-segmented words. Our study adds some steps to the Morfessor segmentation process to consider invalid morphemes and borrowed words from other languages to improve morphological segmentation significantly. Comparing our idea to the results obtained from Morfessor demonstrates its efficiency, leading to more accurate morphology segmentation. This is particularly evident in the case of Turkish, highlighting the potential for further advancements in morpheme segmentation for morphologically rich languages.

1 Introduction

Morphological analysis refers to the examination of word structure. It involves breaking down words into smaller units called morphemes and studying the different morphological rules that can apply to these units (Manning, 1998; Goldsmith, 2010). These rules include inflectional morphology rules, which generate different forms of a word without altering its core meaning, and derivational morphology rules, which create new words based on existing ones by modifying their meanings (Stump, 2001). While both types of rules are important, recent studies have emphasized inflectional morphological analysis more.

Several methods for performing morphological segmentation include supervised, semi-supervised, and unsupervised approaches. Supervised techniques involve using labeled data, such as a collection of previously segmented words, to train a model to recognize similar patterns in new data. However, obtaining large enough datasets to cover all living languages can be prohibitively resource-intensive, requiring significant manual labor and expertise. As a result, many researchers instead opt for unsupervised techniques, which rely solely on raw text data gathered from diverse sources like news articles, movie subtitles, or online content (Virpioja et al., 2011).

Another common technique for morphological segmentation is rule-based approaches, which utilize manually crafted rules to identify morpheme boundaries within words (Narasimhan, 2014). Although effective when implemented correctly, creating and maintaining these rules can be costly and time-consuming, especially for less commonly spoken languages. Consequently, only some practical applications employ this method.

When applying morphological segmentation, it is crucial to consider how morphemes are positioned within language units, as this can vary widely across different languages. For instance, specific languages may place prefixes before roots, while others might use suffixes after roots. Moreover, the complexity of morphological segmentation can pose additional challenges, particularly when creating annotated data, which tends to be expensive and time-consuming. To address this issue, our proposed solution involves developing a comprehensive model capable of handling multiple languages simultaneously. Specifically, we will focus on improving morphological segmentation performance for agglutinative languages such as Finnish and Turkish (Durrant, 2013), which exhibit high levels of morphological complexity.

To accomplish this goal, our project will leverage freely available word lists and perform extensive preprocessing steps to ensure consistency and accuracy. Preprocessing will entail converting each dataset into individual lines containing single words, ensuring uniformity in letter casing and font styles, and eliminating extraneous elements such as punctuation marks, numerals, and duplicated entries. Once completed, we will apply two distinct algorithms to segment the processed data. Our pri-

Proceedings of the The Seventh Workshop on Technologies for Machine Translation of Low-Resource Languages (LoResMT 2024), pages 85–93

mary algorithm will be Morfessor, a probabilistic model explicitly designed for morphological segmentation tasks. We anticipate that evaluating and refining the resulting segmentations through targeted modifications will yield significant improvements in overall performance.

2 Review of Literature

The supervised approach of morphological segmentation uses word information such as part-of-speech (POS) tags, morphological rules, morpheme dictionaries, etc. The unsupervised approach exploits morphemes from a raw corpus (Arabsorkhi and Shamsfard, 2006). This approach has been studied in different languages. There are some mathematical frameworks for modeling methodologies:

- Maximum Likelihood (ML)
- Probabilistic Maximum a Posteriori (MAP)
- Finite state automata (FSA)

MAP modeling is based on the Minimum Description Length (MDL) principle, which considers accuracy and model complexity. An FSA can specify the various word forms (Creutz and Lagus, 2007).

The first Turkish morphology analyzer is Oflazer's model. It is implemented using a PC-KIMMO environment (Oflazer, 1994), which is a computational approach for two-level analyzers (Antworth and McConnell, 1998) and addressing:

- Various forms of words as stated by inflections and derivations
- The dictionary's inability to store up all inflected or derived forms of a word

Külekcỳ and Özkan (2001) proposed a model for dealing with word segmentation. Despite being suggested for Turkish, this model can be used for other languages. In contrast to Oflazer's model, the united stream of characters is used in this model. As a result, to detect segmented morphemes, this model should consider the root and achievable boundaries (Külekcỳ and Özkan, 2001).

Zemberek is an open-source framework for Turkic languages using Latin script that includes language structure information and NLP operations. The standard morphological parser identifies the root and potential suffixes (Akın and Akın, 2007).

Another morphological analyzer is TRmorph, a two-level and rule-based analyzer proposed by Cöltekin (2010). It uses the Stuttgart Finite State Transducer (SFST) tools and consists of 3 major components: a finite state machine (FSA), a set of two-level rules, and a lexicon that keeps the class of root and some lexical irregularities (Cöltekin, 2010). This lexicon is hand-made to be an error-free lexicon created during implementation. Based on morpho-phonological considerations, TRmorph considers morpheme alternations. Morpho-phonological alternation is dependent on phonological alternation. Therefore, Turkish vowels and consonants can be dismissed, reproduced, or revised to the closest harmonic letter. For example:

- ben (I) → ben + ((y)) a → bana (for me)
- hak (right) → hak + ((n)) ı → hakkı (his right)

Spoken Turkish follows a two-level rule system for phonetics, while morphotactics is encoded as finite state machines for word categories like verbs and nouns (Oflazer, 1994). Furthermore, Turkish words have two classes in the Morphotactics part: nominal and verbal. All categories except verbs are in nominal class. Nominal morphotactic is simple. Instead, verbal morphotactics is complicated and has many exceptions. So, both morphotactics should be used simultaneously. This analyzer has been updated and uses the Foma FST compiler, a C compiler, and converts the input to the FST, and the lexicon of TRmorp is a raw text file precisely the prior version (Cöltekin, 2010).

In Turkish, adding morphemes to a word's root or stem can change it from a noun to a verb or vice versa. These morphemes can also create adverbial constructs. This language has exceptions, and Persian, Arabic, and other foreign entered words are considered one of them. The morphological analyzer, used by Şahin et al. (2013), is a two-level analyzer with over 49321 entries, arranged under 14 parts of speech. Their model uses flag diacritics. Flag diacritics' main usage goal is adding a small quantity of memory to the finite state machine during the generation and analysis steps at run-time. If we do not have this, state transitions depend only on the current state and input symbols (Şahin et al., 2013). This model is used in ITU Turkish NLP Web Service[1] consists of different elements such as Tokenizer, Normalizer, Morphology analyzer, Morphology disambiguation, et cetera. In ITU, Components are categorized under four

[1] http://tools.nlp.itu.edu.tr

groups: preprocessing, morphological processing, multiword expression handling, and syntactic processing (Eryiğit, 2014). In ITU, the morphological layer uses a rule-based analyzer that is proposed by Şahin et al. (2013) as well as HFST-Helsinki FST proposed by Lindén et al. (2009) and a hybrid morphological disambiguator (Eryiğit, 2014).

Besides Turkish-specified morphological analyzers, some unsupervised methods can be used for languages without exception. Morfessor is used in our project in this way.

Creutz and Lagus (2002) proposed the main idea of Morfessor in 2002, and they improved their model and named it Morfessor. Morfessor is an unsupervised generative probabilistic model for predicting morphological segmentation. The first version of it is named Morfessor Baseline (Creutz and Lagus, 2007). Then, the idea expanded and introduced other versions, such as Morfessor 1.0, Morfessor FlatCat, Allomorfessor, and Morfessor 2.0 (Creutz and Lagus, 2002; Creutz, 2003; Creutz and Lagus, 2004, 2005). However, the Baseline version is still popular as a morphological analyzer even though other versions have improved the results. Morfessor is well suited for rich morphology languages such as Finnish, Estonian, and German. It uses a corpus of unannotated text as input and produces a segmentation of words observed in the text as its output. Unlike most morphological models, Morfessor is not restricted to the count of morphemes. Morphological analyzers must be built for each language, but Morfessor is a general model for unsupervised and semi-supervised morphological segmentation (Creutz and Lagus, 2007, 2002; Creutz, 2003; Creutz and Lagus, 2004, 2005).

It has been proved that the weighted function leads to better results. Creutz and Lagus (2005) introduced a semi-supervised model based on the Morfessor Baseline. The semi-supervised approach is an excellent way for humans to prepare annotated data manually since data is expensive and complicated to obtain. An important question, however, is how many annotated words are required. Kohonen et al. have proved that 100 manually segmented words is enough and can improve output quality (Creutz and Lagus, 2005).

The algorithm processes all combinations of training data in one cycle. For each combination, it checks every possible two-part division and picks the one with the lowest cost. The cost can decrease or stay the same at each step, which means the algorithm will eventually stop working when the cost drops below a certain level. There is also a way to use this algorithm to decode messages, which involves finding the best division for a new message without changing the program's settings (Creutz and Lagus, 2005). A newer version, Morfessor 2.0, has additional features that allow it to divide messages even when it does not have all the information it needs (Kohonen et al., 2010).

The impact of the smaller size of the annotated data on the cost function is insignificant compared to the likelihood of the unannotated data. Therefore, additional weighting parameters should be included in the annotated data to avoid the adverse effects of annotations on the cost function (Creutz and Lagus, 2007).

The second feature of Morfessor 2.0 is an online and batch training mode. So, the model needs to know how much the final size of training data is, and it has access to only one word of training data at a time. Morfessor 2.0 can skip analyzed compounds and constructions randomly since the variant compounds can be found in the current text. Morfessor 2.0 produces the n-best segmentation of multiple generated segmentations for every compound via the Viterbi algorithm. This feature lets the algorithm extract the most conceivable segmentation for a compound (Smit et al., 2014).

In other studies, Vania and Lopez (2017), and Haghdoost et al. (2019) investigated the effects of different approaches to breaking down words into smaller units for language modeling purposes (Haghdoost et al., 2019, 2020).

Vania and Lopez (2017) examined ten languages with diverse structural properties and trained word-level language models while adding granular information about the constituent parts of each word throughout the training process. After comparing several segmentation techniques, they concluded that character-based models produced superior outcomes overall. Moreover, rule-based approaches tailored to each language's distinct characteristics significantly outperformed alternative partitioning strategies (Vania and Lopez, 2017).

Meanwhile, Haghdoost et al. explored the utility of Morfessor—both supervised and unsupervised variants—for generating morphological networks in Persian and Turkish. Specifically, they determined that the supervised approach was preferable for their objectives, enabling them to incorporate newly segmented words into their lexicon (Haghdoost et al., 2019). Furthermore, they employed both Morfessor versions to establish a Turkish mor-

phological network, thereby revealing relationships among segmented words through a tree-like framework similar to manual segmentation (Haghdoost et al., 2020).

Additionally, research on 50 distinct languages revealed disparities in the difficulty levels associated with crafting language models due to fluctuations in grammatical architectures (Gerz et al., 2018). Prior work demonstrated that Morfessor mitigates morphological impacts for numerous languages, and morphologically driven partitions enhance cross-lingual language modeling (Creutz and Lagus, 2007; Park et al., 2021).

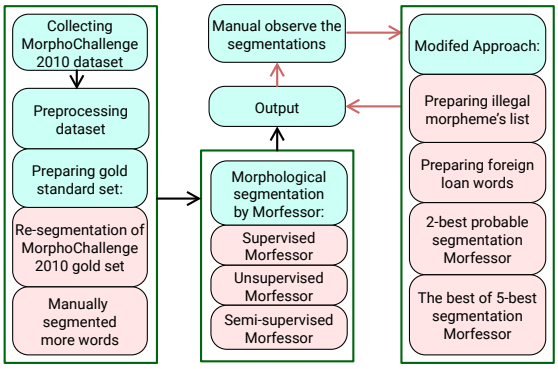

Figure 1: Our approach for Turkish morphological segmentation

3 Our Experiment

Our project aims to accurately segment morphemes using Morfessor, as natural language word lists are time-consuming and labor-intensive to create manually. We address the challenge of creating raw and segmented datasets by using the MorphoChallenge 2010 Turkish dataset[2] as input to Morfessor. Our contributions include gold-segmented words, invalid morphemes (i.e., morphemes not included in the language), and a list of entered foreign words (borrowing words from other languages). Preparing Datasets is not the only part of our work. We use them in different types of Morfessor: supervised, semi-supervised, and unsupervised. Then, we observe the behavior of Morfessor in the segmentation process and according to this, we add some steps to the segmentation process to enhance the performance of Morfessor that will be discussed completely. To put it in a nutshell, our modified process of Morfessor does not let the entered foreign words become separated. It also uses an invalid morpheme dataset to prevent their breaking down from the words. Also, we produce the second and best probable segmentation by Morfessor. All of these experiments are done with the same input dataset- are evaluated and compared with each other. Let's dive into the details in the following subsections.

3.1 Datasets

We use four datasets for our research:

1. word list file: This file contains 617,298 unique Turkish words with their frequency of occurrence. However, as we were interested in finding roots and morphemes, and all types of morphemes are essential, all words were equally treated. Doing so allows us to gain insights into the language's internal workings and better understand how meanings are constructed. Therefore, we removed the frequency of the words and numbers, punctuation marks, and other unnecessary items in the pre-processing phase.

2. Gold standard file: We used the MorphoChallenge 2010 gold standard file consisting of 1000 words, but with some changes. We reviewed this gold standard set and noticed that some words can be more re-segmented according to the language rules to create derived words for improving the accuracy of our evaluation. We also randomly selected 1000 unique words from the word list file, segmented them manually, and merged these new segmentations with the corrected gold standard file. This resulted in a dataset of 2000 segmented words without overlap between these samples, of which 20% was used for testing, and the remainder was used for training or addition to the word list file, following our semi-supervised approach.

3. Borrowed words: This dataset contains foreign words that have entered Turkish throughout history and culture. We manually gathered this dataset from websites and online Turkish texts. The foreign word dataset contains 5553 French, 1,523 Persian, 1,188 English, and 626 Arabic words that entered Turkish.

4. Invalid morphemes: This dataset contains a small number of letter combinations recognized as morphemes by Morfessor but not actual morphemes.

[2] http://morpho.aalto.fi/events/morphochallenge2010/datasets.shtml

3.2 Data Processing

We used the word list file as input to Morfessor in three ways: supervised, unsupervised, and semi-supervised. The output of all approaches was the first probable segmentation of data. We also produced the two most probable segmentations for each word to observe the segmentation process in the next probable segmentations.

3.3 Applying Morfessor

Experimentation with Morfessor progressed in three primary manners: supervised, semi-supervised, and unsupervised Morfessor. Each experimental setup featured varying degrees of human intervention.

- **Supervised Morfessor:** We leveraged 1,600 segmented words as a training set and reserved 400 words for testing purposes, sourcing both sets randomly from the pool of gold standard segmented words. This configuration remained consistent throughout all the approaches tested.

- **Semi-supervised Morfessor:** Under this setting, we combined 1,600 segmented words along with 8,400 unsegmented words to formulate the baseline input dataset. Employing a hybrid mix of segmented and unsegmented words enabled the algorithm to learn from a broader array of examples, thereby enhancing its adaptability towards segmentation tasks.

- **Unsupervised Morfessor:** Lastly, we presented the algorithm with a random assortment comprising 10,000 unsegmented words extracted from the word lists. Completely devoid of any prior guidance, the unsupervised version of Morfessor embarked upon discovering meaningful segmentations autonomously. It should be mentioned that the words are the same in the datasets we use in each experiment in order to have fair results.

These divergent strategies allowed us to gauge the efficacy of Morfessor across a spectrum of scenarios, ranging from heavily guided environments to entirely self-reliant conditions. Ultimately, understanding the nuances of Morfessor's performance under various circumstances shall aid us in optimizing its application for real-world problems.

3.4 Invalid Morpheme Handling

We developed a process to handle invalid morphemes in the output of Morfessor. This process aimed to prepare a list of invalid morphemes that cannot be used in Turkish and then to use this list to select the most probable segmentation for each word. The process worked as follows:

- We produced the first two most probable segmentations for each word.

- We selected the second segmentation if there were invalid morphemes in the first segmentation.

- If there were also invalid morphemes in the second segmentation, we selected the most probable candidate that did not contain any invalid morphemes.

- If no valid segmentation existed, we returned the first one.

3.5 Evaluation

During the evaluation process, we consider the segmented portions, the word boundaries, and morphemes. Ensuring precise delineation of morpheme boundaries plays a pivotal role in determining the quality of the segmentation results, alongside taking into consideration the typical morphemes encountered in both the output and gold segmentation files. We used Precision and Recall as the metrics of evaluation to compare our results. For Morpheme segmentation, segmented parts of the word must be evaluated. Additionally to segmented characters, the start and end of a word are crucial to determining a boundary and evaluating the boundary Precision and Recall. The final component of segmentation evaluation is the correct segmentation.

$$Precision = \frac{TruePositive}{TruePositive + FalsePositive}$$

$$Recall = \frac{TruePositive}{TruePositive + FalseNegative}$$

For instance, in the output word "danesh gah" and the gold set "dan esh gah" (daneshgah is a Persian word for university), we have two segmented morphemes in "danesh gah" but the gold segmentation has three segmented morphemes that one of them occurs in the output, i.e. "dan esh gah". Therefore, morpheme precision for this word is $\frac{1}{3}$. We should consider the first and end of the word

for boundary precision. So in the gold set for this word, we have two spaces between three segmentations: first and end (i.e., $2 + 1 + 1$), and in output, we have one space between segmented morphemes plus the first and end of the word (i.e., $1 + 1 + 1$). Therefore, the boundary precision will be $\frac{3}{4}$. Segmentation precision is correct or not, so it will be $\frac{0}{1}$, i.e., 0.

As discussed in 3.3, we used Morfessor to segment our data. This is essential for later evaluation and comparison with Morfessor itself. The Morfessor with semi-supervised learning is effective on the given data, and we will continue to utilize this approach in the subsequent stages of our experiments.

Morfessor- Turkish	BP	MP	SP	BR	MR	SR
Supervised	0.570	0.093	0.037	0.372	0.055	0.037
Unsupervised	0.583	0.130	0.032	0.468	0.097	0.032
Semi-supervised	**0.614**	**0.134**	**0.052**	**0.485**	**0.099**	**0.052**

Table 1: Evaluation of Morfessor on Turkish input (BR: boundary recall, BP: boundary precision, MR: morpheme recall, MR: morpheme precision, SR: segmentation recall, SP: segmentation precision)

4 Results and Observations

Curious about Morfessor's inner workings, we examined Morfessor's output via visual inspection. Starting with a random selection of 200,000 words from the word list, we focused on the semi-supervised Morfessor's behavior. Segmenting 1,600 words from the gold standard file, we allocated 400 words as the evaluation test set. The random words were selected because we only wanted to observe the Morfessor in an unbiased and fair way.

Throughout the segmentation process, Morfessor adeptly identified suffixes and partitioned associated suffix groups into one or multiple components. Nonetheless, it maintained a reasonable count of word fragments. Syllabification played a vital role, enabling efficient segmentation with reduced syllable counts.

Morfessor categorized words lacking suffixes into two classes: monosyllabic and polysyllabic. Monosyllabic words remained undivided, whereas polysyllabic counterparts experienced arbitrary breakage into smaller pieces.

Segmenting words attached with suffixes, Morfessor isolated the base word and either singular or compound suffixes, subject to word length and contextual factors. Diminutive suffixes could face division or remain cohesive according to the prevailing situation. Overall, Morfessor's functional design shed light on its competence in achieving satisfactory segmentation outcomes.

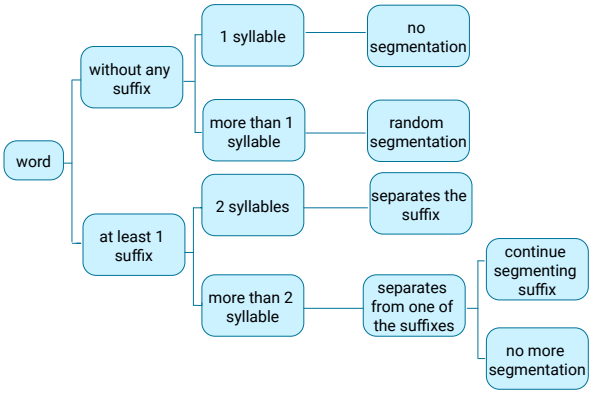

Figure 2: How Morfessor separates the words in general

As a result, the Morfessor has its logic to separate words, sometimes leading to errors. For example, the suffix "ler" indicates the plural form of the word and should be separated from the root word. However, when this suffix combines with another suffix like "ler i", Morfessor incorrectly separates the word into the root word and "leri". Alternatively, "roman", a French word that means novel, is broken down due to its last syllable "an", which is precisely similar to the "an" suffix in Turkish. Morfessor separates words more or halts them if there is more than one legal and invalid suffix in a more than two-syllable word structure. Therefore, foreign word splitting is a challenge that should be solved.

Based on our observations, we can derive additional insights:

1. Morfessor does not segment words into the smallest possible units. We, therefore, choose the second segmentation in the first concept.

2. The second concept is to select the best segmentation from Morfessor's five possible outputs. We select the segmentation that does not contain invalid suffixes.

3. The third concept is to keep foreign-borrowed words intact. We implement these concepts by randomly selecting 9000 words from the Turkish dataset and dividing the gold standard file into 400 words for the test set and 1600 for the training set. Morfessor is also trained on these 1600 words. Therefore, our input dataset contains 10600 words.

As we see in Table 1, the semi-supervised Morfessor has the best result. Therefore, we chose it as the benchmark for our further experience to examine our idea and evaluate it. The results are in Table 2.

Semi-supervised Morfessor	BP	MP	SP	BR	MR	SR
1-best segmentation	0.576	0.114	0.052	0.444	0.082	0.052
2-best segmentation	0.579	0.117	0.055	0.441	0.083	0.055
The invalid morphemes segmentation	0.579	0.118	0.052	0.440	0.084	0.052
Considering borrowed words	**0.627**	**0.158**	**0.140**	**0.531**	**0.128**	**0.140**

Table 2: Results of the most probable best segmentation and considering borrowing words in Turkish (BR: boundary recall, BP: boundary precision, MR: morpheme recall, MR: morpheme precision, SR: segmentation recall, SP: segmentation precision)

Morfessor generates the most likely segmentation as the "1-best". To better understand the impact of other possible segmentations, we choose the "2-best" segmentation. The table shows that precision improves slightly while recall decreases slightly. Another option is to select the best of the "5-best" segmentations while accounting for invalid morphemes. If an invalid morpheme is detected, Morfessor will choose the subsequent segmentation that does not contain that morpheme. If no segmentations are missing, the algorithm will return the first segmentation as the best. Because the results did not improve significantly, we tried another idea. In this case, we used the foreign borrowed words we prepared. We will check to see if Morfessor produced any borrowed words. If so, Morfessor should combine the segments, as these are prohibited morphemes.

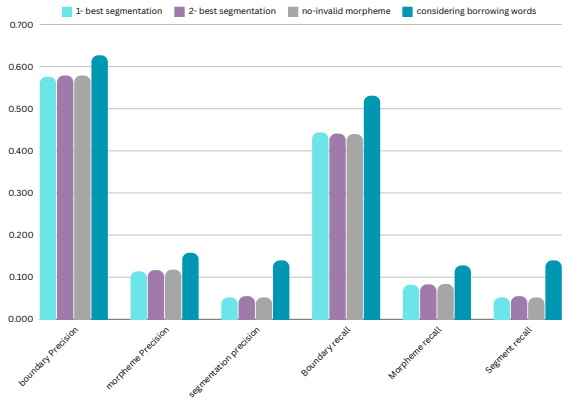

Figure 3: Illustration of the observed ideas

The graph shows that considering borrowed words significantly improved Morfessor's performance. This suggests that having a dataset of borrowed words can improve Morfessor's accuracy and obtain a more detailed segmentation list.

5 Conclusion

In this study, we decided to focus on Morfessor 2.0, a powerful probabilistic tool designed for morphological segmentation specifically tailored for the Turkish language. Our objective was clear: achieve accurate morpheme segmentation in Turkish and discover methods to improve Morfessor's overall performance. We started by utilizing the MorphoChallenge 2010 Turkish dataset, then expanded our sources by gathering extra data comprising borrowed words and even invalid morphemes.

Taking a closer look at different methodologies, our research involved three main approaches—supervised, semi-supervised, and unsupervised learning applied to Morfessor. When measuring Morfessor's effectiveness, we relied on six essential evaluation metrics: boundary recall, boundary precision, morpheme recall, morpheme precision, segmentation recall, and segmentation precision. After thorough analysis, one particular technique stood out among the rest—the semi-supervised approach demonstrated superior accuracy compared to the others.

While exploring how Morfessor functions, we noticed some remarkable abilities; primarily, it excels in recognizing suffixes and breaking them down into smaller sets. Crucially, though, it avoids excessive fragmentation during this breakdown process. Despite those strengths, however, there were still difficulties faced in segmenting foreign terms. Addressing this challenge head-on, we suggested innovative strategies to boost Morfessor's capacity to handle foreign vocabulary better. These creative solutions entailed examining alternative segmentations (second-best), discarding faulty morphemes, and incorporating borrowed phrases from various languages. Upon merging incorrect morphemes and foreign loans throughout the segmentation procedure, we witnessed notable progress in both precision and recall ratios related to Turkish. For interested readers, you can find all relevant experiments and corresponding outputs on our project's GitHub page [3].

Wrapping up our investigation, we believe our work constitutes a major leap forward in crafting a far-reaching segmentation algorithm equipped to skillfully tackle the labyrinthine nuances found in not only Turkish but also other highly inflected languages. As part of our future plans, we intend

[3] https://github.com/Soheila-Behrooznia/TurkishMorphologySegmentation

to spotlight and incorporate even more extensive catalogs of foreign terminology present in Turkish, ultimately leading to enhanced precision concerning segmented word directories.

Acknowledgements

This work has been partially supported by Charles University Research Centre program No. 24/SSH/009.

We would like to thank three anonymous reviewers for their very insightful feedback.

References

Ahmet Afsin Akın and Mehmet Dündar Akın. 2007. Zemberek, an open source nlp framework for turkic languages. *Structure*, 10(2007):1–5.

Evan Antworth and S McConnell. 1998. Pc-kimmo reference manual. *A two-level processor for morphological analysis, version*, 2(0).

Mohsen Arabsorkhi and Mehrnoush Shamsfard. 2006. Unsupervised discovery of persian morphemes. In *Demonstrations*, pages 175–178.

Cagri Cöltekin. 2010. A freely available morphological analyzer for turkish. In *LREC*, volume 2, pages 19–28.

Mathias Creutz. 2003. Unsupervised segmentation of words using prior distributions of morph length and frequency. In *Annual Meeting of the Association for Computational Linguistics*, pages 280–287. ACL.

Mathias Creutz and Krista Lagus. 2002. Unsupervised discovery of morphemes. *arXiv preprint cs/0205057*.

Mathias Creutz and Krista Lagus. 2005. 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)*, volume 1, pages 51–59.

Mathias Creutz and Krista Lagus. 2007. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing (TSLP)*, 4(1):1–34.

Mathias Johan Philip Creutz and Krista Hannele Lagus. 2004. Induction of a simple morphology for highly-inflecting languages. In *7th Meeting of the ACL Special Interest Group in Computational Phonology (SIGPHON)*, pages 43–51.

Philip Durrant. 2013. Formulaicity in an agglutinating language: The case of turkish. *Corpus Linguistics and Linguistic Theory*, 9(1):1–38.

Gülşen Eryiğit. 2014. Itu turkish nlp web service. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 1–4.

Daniela Gerz, Ivan Vulić, Edoardo Ponti, Jason Naradowsky, Roi Reichart, and Anna Korhonen. 2018. Language modeling for morphologically rich languages: Character-aware modeling for word-level prediction. *Transactions of the Association for Computational Linguistics*, 6:451–465.

John A Goldsmith. 2010. Segmentation and morphology. *The handbook of computational linguistics and natural language processing*, pages 364–393.

Hamid Haghdoost, Ebrahim Ansari, Zdenek Žabokrtský, Mahshid Nikravesh, and Mohammad Mahmoudi. 2020. Morphological networks for persian and turkish: What can be induced from morpheme segmentation? *Prague Bull. Math. Linguistics*, 115:105–128.

Hamid Haghdoost, Ebrahim Ansari, Zdeněk Žabokrtský, and Mahshid Nikravesh. 2019. Building a morphological network for persian on top of a morpheme-segmented lexicon. In *Proceedings of the Second International Workshop on Resources and Tools for Derivational Morphology*, pages 91–100.

Oskar Kohonen, Sami Virpioja, and Krista Lagus. 2010. Semi-supervised learning of concatenative morphology. In *Proceedings of the 11th Meeting of the ACL Special Interest Group on Computational Morphology and Phonology*, pages 78–86.

M Oguzhan Külekcỳ and Mehmed Özkan. 2001. Turkish word segmentation using morphological analyzer. In *Seventh European Conference on Speech Communication and Technology*.

Krister Lindén, Miikka Silfverberg, and Tommi Pirinen. 2009. Hfst tools for morphology–an efficient open-source package for construction of morphological analyzers. In *State of the Art in Computational Morphology: Workshop on Systems and Frameworks for Computational Morphology, SFCM 2009, Zurich, Switzerland, September 4, 2009. Proceedings*, pages 28–47. Springer.

Christopher D Manning. 1998. The segmentation problem in morphology learning. In *New Methods in Language Processing and Computational Natural Language Learning*.

Karthik Rajagopal Narasimhan. 2014. *Morphological segmentation: an unsupervised method and application to Keyword Spotting*. Ph.D. thesis, Massachusetts Institute of Technology.

Kemal Oflazer. 1994. Two-level description of turkish morphology. *Literary and linguistic computing*, 9(2):137–148.

Hyunji Hayley Park, Katherine J Zhang, Coleman Haley, Kenneth Steimel, Han Liu, and Lane Schwartz. 2021. Morphology matters: A multilingual language

modeling analysis. *Transactions of the Association for Computational Linguistics*, 9:261–276.

Muhammet Şahin, Umut Sulubacak, and Gülşen Eryiğit. 2013. Redefinition of turkish morphology using flag diacritics. In *The 10th Symposium on Natural Language Processing*. Thammasat University.

Peter Smit, Sami Virpioja, Stig-Arne Grönroos, Mikko Kurimo, et al. 2014. Morfessor 2.0: Toolkit for statistical morphological segmentation. In *The 14th Conference of the European Chapter of the Association for Computational Linguistics (EACL), Gothenburg, Sweden, April 26-30, 2014*. Aalto University.

Gregory T Stump. 2001. *Inflectional morphology: A theory of paradigm structure*, volume 93. Cambridge University Press.

Clara Vania and Adam Lopez. 2017. From characters to words to in between: Do we capture morphology? *arXiv preprint arXiv:1704.08352*.

Sami Virpioja, Ville T Turunen, Sebastian Spiegler, Oskar Kohonen, and Mikko Kurimo. 2011. Empirical comparison of evaluation methods for unsupervised learning of morphology. *Traitement Automatique des Langues*, 52(2):45–90.