# NEURAL NAMED ENTITY RECOGNITION FOR MORPHOLOGICALLY RICH LANGUAGES

by

Onur Güngör

B.S., Computer Engineering, Boğaziçi University, 2006

M.S., Computer Engineering, Boğaziçi University, 2009

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy

Graduate Program in Computer Engineering

Boğaziçi University

2021

# ACKNOWLEDGEMENTS

First and foremost, I would like to express my gratitude to my advisors Dr. Suzan Üsküdarlı and Prof. Tunga Güngör. Besides being my advisors in this thesis, I have taken several of their courses, received valuable advice, and used their wisdom to shape my academic career on many occasions for nearly all my academic life.

This thesis is an important milestone of my learning journey which will continue for all my life. I have become and will remain a student, a scholar, a follower of scientific method. Most importantly I will be working to spread the knowledge and the personality traits that make up a scholar to other curious people.

When I look through this journey, some people stand out among others. I would like to thank Prof. Taylan Cemgil for providing me with a strong foundation to understand statistics and machine learning and advising me in my early thesis work. I would like to thank Prof. Haluk Bingöl for lecturing us about the importance of scientific rigor and asking the right questions as early as my undergraduate years.

I would like to thank my thesis committee members Assoc. Prof. Arzucan Özgür and Prof. İlker Birbil for their valuable advice that always helped me to correct course. I would also like to thank Assist. Prof. Fatma Başak Aydemir and Assist. Prof. Reyyan Yeniterzi for kindly accepting to serve in my thesis jury and their comments.

I must also acknowledge the welcoming atmosphere that nurtures free speech and freedom of taking initiatives in the department of computer engineering. I witnessed productive collaborations and efficient work done many times due to this atmosphere. I want to thank wonderful graduate students and professors who keep this atmosphere alive. But among them, although it was very hard to pick, I would like to thank some of them. Ahmet Yıldırım was an ideal labmate whom I had the chance to discuss a wide range of topics and have good time. Arda Çelebi is one of my go-to people for natural language processing related questions who helped me a lot whenever I wanted

to test an idea for feasibility. I'll be remembering the fellowship of you when I recall my PhD years. It would be a pity not to mention the heartfelt friendship of the TABI Lab crew, thank you all.

Lastly, I want to thank my family for bearing the burden of this long and life-changing experience with me. My deepest gratitude goes to my spouse and comrade Ekin Sönmez, this would not be possible without her support and encouragement. I would also thank my dear mother for laying the groundwork for what is Onur now, and for her patience to wait more than 10 years to attend a graduation ceremony which I hope we could arrange face-to-face despite the COVID-19 pandemic. They say that it takes a village to raise a child, and so did in my case. I want to thank my father for being a role model for many things, my comrades for helping me to thrive and grow roots instead of being a flâneur, and Sercan Kabakcı for being a sensible thinker and a long-lasting friend.

Through the first four years of my PhD, I was supported by the Scientific and Technical Research Council of Turkey (TUBİTAK) BIDEB-2211 Scholarship program.

# ABSTRACT

## NEURAL NAMED ENTITY RECOGNITION FOR MORPHOLOGICALLY RICH LANGUAGES

Named entity recognition (NER) is an important task in natural language processing (NLP). Until the revival of neural network based models for NLP, NER taggers employed traditional machine learning approaches or finite-state transducers to detect the entities in a given sentence. Neural models improved the state-of-the-art performance with sequence-based models and word embeddings. These approaches neglect the morphological information embedded in the surface forms of the words. In this thesis, we introduce two NER taggers that utilize such information, which we show to be significant for morphologically rich languages. Using these taggers, we improve the state-of-the-art performance levels for Turkish, Czech, Hungarian, Finnish, and Spanish. The ablation studies show that these improvements result from the inclusion of morphological information. We also show that it is possible for the neural network to also learn how to disambiguate morphological analyses, thereby, eliminating the dependence on external morphological disambiguators that are not always available. In the second part of this thesis, we propose a model agnostic approach for explaining any sequence-based NLP task by extending a well-known feature-attribution method. We assess the plausibility of the explanations for our NER tagger for Turkish and Finnish through several novel experiments.

# ÖZET

## BİÇİMBİLİMSEL AÇIDAN ZENGİN DİLLERDE SİNİR AĞI TABANLI VARLIK İSMİ TANIMA

Varlık ismi tanıma (VİT), doğal dil işleme (DDİ) alanınındaki önemli bir görevdir. VİT etiketleyicileri verili bir cümledeki varlıkları etiketlemek için sinir ağı tabanlı modellerin yeniden doğuşuna kadar geleneksel yapay öğrenme yöntemleri veya sınırlı durumlu dönüştürücüleri kullanmaktaydı. Dizi tabanlı modelleri veya sözcük temsillerini kullanan sinir ağları o zamana kadar elde edilmiş en iyi başarımları ilerletti. Bu yaklaşımlar sözcüklerin yüzey biçimlerindeki biçimbilimsel anlam ifade eden bilgiyi görmezden gelmiştir. Bu tezde, biçimbilimsel bilgiyi kullanan iki VİT etiketleyicisi sunulmakta ve bu tür bilginin kullanılmasının biçimbilimsel açıdan zengin dillerdeki başarıyı önemli derecede artırdığı gösterilmektedir. Bu etiketleyiciler kullanılarak Türkçe, Çekçe, Macarca, Fince ve İspanyolca VİT görevinde o zamana kadar elde edilmiş en iyi başarımlar ilerletilmiştir. Modelin çeşitli kesimlerini etkin veya devredışı kılarak yaptığımız deneylerle bu ilerlemenin biçimbilimsel bilginin modele dahil edilmesinden kaynaklandığı gösterilmiştir. Bunlara ek olarak, olası tüm biçimbilimsel çözümlemelerden doğru olanı seçme işinin her zaman elde edilmesi mümkün olmayan harici çözümleyiciler kullanmadan sinir ağının bir parçası olarak yapılabileceği gösterilmiştir. Tezin ikinci kısmında, bilinen bir öznitelik ilişkilendirme yöntemi temel alınarak herhangi bir model türüne özgü olmayan bir açıklama getirme yöntemi geliştirilmiştir. Bu yöntemin ürettiği açıklamaların ikna ediciliği ilk kısımda geliştirilen VİT etiketleyiciler kullanılarak çeşitli özgün deneylerle gösterilmiştir.

# TABLE OF CONTENTS

ACKNOWLEDGEMENTS . . . iii
ABSTRACT . . . v
ÖZET . . . vi
LIST OF FIGURES . . . x
LIST OF TABLES . . . xiii
LIST OF SYMBOLS . . . xvi
LIST OF ACRONYMS/ABBREVIATIONS . . . xvii
1. INTRODUCTION . . . 1
1.1. Contributions . . . 5
1.2. Thesis outline . . . 7
2. RELATED WORK . . . 8
2.1. Named entity recognition . . . 8
2.2. Morphological analysis and disambiguation . . . 10
2.3. Joint modeling . . . 11
2.4. Interpretability . . . 12
2.5. IOB and IOBES tagging schema for NER . . . 15
3. MORPHOLOGICAL INFORMATION FOR NAMED ENTITY RECOGNITION . . . 16
3.1. Introduction . . . 16
3.2. Background . . . 17
3.2.1. Morphologically Rich Languages . . . 17
3.2.1.1. Turkish . . . 17
3.2.1.2. Finnish . . . 18
3.2.1.3. Czech . . . 18
3.2.1.4. Hungarian . . . 19
3.2.1.5. Spanish . . . 19
3.2.2. Bidirectional LSTMs . . . 20
3.3. Model . . . 21
3.3.1. Embeddings . . . 24

3.3.2. Morphological embedding configurations . . . 26
3.4. Experiments . . . 28
3.4.1. Training . . . 28
3.4.2. Datasets . . . 29
3.4.3. Results . . . 31
3.5. Conclusions . . . 34
4. NAMED ENTITY RECOGNITION AND MORPHOLOGICAL DISAMBIGUATION WITH A JOINT MODEL . . . 39
4.1. Models . . . 40
4.1.1. NER Model . . . 41
4.1.1.1. Representing words . . . 42
4.1.1.2. External morphological features . . . 42
4.1.2. MD Model . . . 43
4.1.3. Joint model for NER and MD . . . 45
4.1.3.1. Approach 1: Two losses . . . 45
4.1.3.2. Approach 2: Enriching the context vector . . . 45
4.1.3.3. Approach 3: Multilayer and Shortcut Connections . . . 46
4.2. Data . . . 46
4.2.1. Training . . . 47
4.3. Experiments and results . . . 48
4.4. Conclusions . . . 50
5. EXPLAINING THE PREDICTIONS OF SEQUENTIAL TAGGERS . . . 53
5.1. Introduction . . . 53
5.2. The base explanation method: LIME . . . 56
5.3. Explaining sequence-based NLP tasks . . . 57
5.3.1. Defining NLP tasks . . . 58
5.3.2. EXSEQREG: Explaining sequence-based NLP tasks with regions . . . 59
5.3.3. Perturbation . . . 62
5.3.4. Calculating probabilities . . . 64
5.3.5. Computing importance values . . . 65
5.4. Analysis . . . 68

5.4.1. Results . . . 68
5.4.2. Metrics . . . 70
5.4.3. Using standardized mean importance values . . . 71
5.4.3.1. Finnish . . . 72
5.4.3.2. Turkish . . . 76
5.4.4. Importance distributions of morphological tags . . . 77
5.4.5. Importance of morphological tags across the entity tags . . . 77
5.4.6. Quantitative validation using mutual information . . . 78
5.4.7. The impact of the absence of a morphological tag . . . 78
5.5. Conclusions . . . 80
6. DISCUSSION AND FUTURE WORK . . . 91
6.1. NER for Morphologically Rich Languages . . . 91
6.2. Explaining the predictions of NLP systems . . . 94
7. CONCLUSION . . . 98
REFERENCES . . . 99
APPENDIX A: ADDITIONAL TABLES REGARDING CHAPTER 5 . . . 117

# LIST OF FIGURES

1.1 An excerpt from an opinion article published at *The Atlantic*. The NER system used while creating this illustration was targeting organization, person and location names, each marked with 'ORG', 'PERSON', and 'LOCATION' labels. . . . . 2

1.2 An explanation regarding the prediction of `PRODUCT` for the first three words of a Finnish sentence. Each bar is associated with a specific morphological feature. The height of a bar indicates the importance factor while the direction indicates the effect to be positive or negative. . . . . 5

3.1 NER sample with IOB tagging scheme. . . . . 22

3.2 Processing a Turkish sentence with the proposed model which can be translated as 'UN officer Rolf Ekéus was in Istanbul'. Words are represented with a fixed length vector that combines the word, character, and morphological embeddings (see Section 3.3.2). . . . 23

3.3 The representation of the Turkish word "evlerinde" according to the proposed model featuring all three components: word, character and morphological embeddings. . . . . 26

4.1 The Bi-LSTM model that generates word representations. Word or root surface forms and morphological tag sequences are treated using this architecture. The input sequence $(e_1, e_2, \cdots, e_{n-1}, e_n)$ can either be the character sequence of the whole word or the root of the word, or the tags in the morphological tag sequence. RELU unit is only active for roots and morphological tag sequences. . . 42

4.2 Morphological analysis of Turkish word 'evlerinde' (in his/her houses) and its corresponding transformation. . . . . 43

4.3 Our basic models: NER and MD. The portions of the model which are only active either for NER or MD models are indicated with dashed lines. The symbol ⊠ represents the selection of $\mathtt{ma}_{ij*}$. . . . 44

4.4 Our joint models: (a) JOINT1 and JOINT2 models (b) J_MULTI model. The symbol ☒ represents the selection of $\texttt{ma}_{ij^*}$. . . . . 51

4.5 Graphical user interface of the tool that was developed to comb through the dataset to identify most frequent erroneous analyses and replace them with sensible ones. . . . . 52

5.1 A Turkish sentence (translated as "She/he had not played at the Ali Sami Yen Stadium yet.") with one named entity tag and the possible morphological analyses of the tokens in the named entity. 55

5.2 Relations between processes, models, input, and output in the scope of our explanation method. . . . . 59

5.3 Explanation of a NER tagger's prediction where $N_t = 7, N_f = 7, N_y = 7, N_o = 0$. $X_i$ is the $i$th sentence. Only first seven tokens of the sentence are shown. They translate to "Jeff Barr, the chief evangelist of Amazon Web Services, says ...". Region $r_{i1}$ is labeled as a 'PRO' entity tag. All possible morphological analyses of first, second, and third tokens ($t$) in region $r_{i1}$ that are used to form feature sets $F_{it}$ are shown. $\mathbb{F}_{i1}$ is the set of morphological tags resulting from the union of all $F_{it}$ in region $r_{i1}$. $\mathbf{e}_{i1}^{\texttt{PRO}\bigstar}$ is the resulting explanation vector. . . . . 81

5.4 The change in $\mathbb{F}_{i1}$ corresponding to every perturbed sentence $\pi_{i1}^{r}$ is depicted. Crossed out morphological tags are removed from $\mathbb{F}_{i1}$. Morphological tags which are not already present are not shown. . 82

5.5 A NER task for a Turkish sentence (meaning "Ali Sami Yen Stadium had not yet been relocated."). The token-level tag predictions for each position are shown with scores which are denoted with $s_{t,o}$. The correct sequence of the token-level named entity tags are marked in red. Single token entities are not applicable to multiple token entities, thus their scores are N/A. . . . . 82

5.6 The explanation method for sequence-based NLP tasks. `predict` function relies on the model to obtain the probabilities for each label $k$. . . . . 83

5.7 A subset of the graph that is formed based on the FiNER rules. It shows the nodes and the connections that lie between the 'Location' top rule and the morphological tags reachable using the rules. We only show the subgraph reachable from 'LocGeneral' in finer detail, while replacing the other subgraphs with triangles. Morphological tag nodes are represented with a rectangle. Circles represent the internal rules. . . . . 85

5.8 The histograms of importance values for various entity tag and morphological tag combinations. Darker colors indicate higher frequencies. Calculated over the training corpus in Finnish. All histograms in the figures share the same bin edges. Edge positions are calculated by $-10^{i/10}$ and $10^{i/10}$ for the negative and positive sides, respectively, where $i \in \{-25, \ldots, 13\}$. . . . . 87

# LIST OF TABLES


3.1 Embedding configurations for Turkish, Czech, Hungarian, Finnish and Spanish: `WITH ROOT (WR)`, `WITHOUT ROOT (WOR)`, `WITH ROOT AND AFTER LAST DB (WR_ADB)`, `CHAR`. . . . . 36

3.2 The number of labels for each entity type in Czech and Finnish datasets. . . . . 37

3.3 The performance of the model using various embedding configurations for five languages. . . . . 37

3.4 Comparison of results with state-of-the-art NER results for each language. . . . . 38

4.1 Evaluation of our models for NER performance with our dataset. We report F1-measure results over the test portion of our dataset averaged over 10 replications of the training with the same hyper parameters. . . . . 49

4.2 Evaluation of our models for MD performance. As in the NER evaluation, we report accuracies over the test dataset averaged over 10 replications of the training. . . . . 50

5.1 Selected examples of NLP tasks that can be covered by the method. 60

5.2 Summary of variables used by the explanation method. . . . . 61

5.3 The hyperparameters and region-related statistics for Turkish and Finnish NER datasets. . . . . 69

5.4 The ranks of $\hat{\mu}_k(m)$ for each entity tag in Finnish. The morphological tags that are in the first 8 for at least one entity tag are shown. . . . . 73

5.5 Three FiNER rules are presented in the order of increasing generality. The first one acts on single words. The second applies a rule on a single word and requires the right context to match another rule. The last one is the top rule for the 'Location' tag. Options 1, 2, 3, and 6 can also be seen in Figure 5.7 as they lie on paths that reach a morphological tag while other options do not lead to any. . 84

5.6 Results of the matching between the proposed explanation method and the FiNER tagger for the five most-frequently occurring named entity tags. . . . . 85

5.7 The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown. . . . . 86

5.8 The frequency of ranks that are in the first ten of an entity tag for each morphological tag in Finnish. . . . . 88

5.9 Common Finnish and Turkish morphological tags that are both in $\mathbb{I}_k$ and $\mathbb{J}_k$. . . . . 89

5.10 Comparison between each 'related tag' and 'unrelated tag' sets using F-measure and Recall metrics. . . . . 90

A.1 The ranks of $\hat{\mu}_k(m)$ for each entity tag in Finnish. The morphological tags that are in the first 10 for at least one entity tag are shown. (1/3) . . . . . 118

A.2 The ranks of $\hat{\mu}_k(m)$ for each entity tag in Finnish. The morphological tags that are in the first 10 for at least one entity tag are shown. (2/3) . . . . . 119

A.3 The ranks of $\hat{\mu}_k(m)$ for each entity tag in Finnish. The morphological tags that are in the first 10 for at least one entity tag are shown. (3/3) . . . . . 120

A.4 The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown. (1/5) . . . . . 121

A.5 The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown. (2/5) . . . . . 122

A.6 The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown. (3/5) . . . . . 123

A.7 The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown. (4/5) . . . . . 124

A.8 The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown. (5/5) . . . . . 125

# LIST OF SYMBOLS

| | |
|---|---|
| $\overrightarrow{H}, \overleftarrow{H}$ | The forward and backward cell output matrices of a Bi-LSTM |
| $d_w$ | The number of dimensions of the pretrained word embeddings |
| $d_c$ | The number of dimensions of the Bi-LSTM used to obtain character embeddings |
| $d_m$ | The number of dimensions of the Bi-LSTM used to obtain morphological embeddings |
| $\mathbf{e}_{ij}^{k\bigstar}$ | The vector that explains the prediction of label $k$ for the $j$th region of sentence $i$. It consists of importance values corresponding to each feature. |
| $\mathtt{ma}_i$ | The set of candidate morphological analyses for word $i$ |
| $\mathtt{ma}_{ij*}$ | The disambiguated morphological analysis for word $i$ |
| $\hat{\mu}_k$ | The mean of standardized importance values |
| $\hat{\mathbb{E}}^k(m)$ | The distribution of standardized importance values across the corpus |
| $MI_{k,m}$ | The mutual information gain between a label ($k$) and a feature ($m$) (a NER label and a morphological tag in the use case) |
| $\pi_{ij}$ | The set of perturbed sentences originating from $r_{ij}$ |
| $P_{ij}$ | The matrix of probability differences induced by perturbing region $r_{ij}$ |
| $r_{ij}$ | (i) The root representation vector for word $i$ and analysis $j$ |
| $r_{ij}$ | (ii) (in EXSEQREG) The region $j$ in sentence $i$ |
| $\mathtt{ms}_{ij}$ | The morpheme sequence representation vector for word $i$ and analysis $j$ |
| $\xi_i$ | The tag score vector for position $i$ of a sentence |

# LIST OF ACRONYMS/ABBREVIATIONS

| | |
|---|---|
| Bi-LSTM | Bidirectional Long Short-term Memory |
| CRF | Conditional Random Field |
| `EXT_M_FEAT` | Multi-layered architecture with shortcut connections in Chapter 4 |
| GRU | Gated Recurrent Unit |
| IG | Inflectional Group |
| IOB | A scheme for NER labeling. It consists of the first letters of **I**nside, **O**utside, and **B**eginning (Section 2.5) |
| IOBES | A scheme for NER labeling. It consists of the first letters of **I**nside-**O**utside-**B**eginning-**E**nd-**S**ingle (Section 2.5) |
| JOINT1 | First joint architecture in Chapter 4 |
| JOINT2 | Second joint architecture in Chapter 4 |
| J_MULTI | The most complicated joint architecture with multiple layers and shortcut connections in Chapter 4 |
| LIME | 'Local Interpretable Model-Agnostic Explanations' |
| LSTM | Long Short-term Memory |
| MD | Morphological Disambiguation |
| MRL | Morphologically Rich Language |
| NER | Named Entity Recognition |
| POS | Part-of-Speech |
| RNN | Recurrent Neural Networks |

# 1. INTRODUCTION

Natural language processing (NLP) aims to process human language samples to produce some type of useful output. There are numerous well-known tasks related to NLP, such as finding the entities referred in a given text, known as named entity recognition (NER). Another task aims to answer questions asked by people who are seeking information about a specific subject, which is called question answering. These tasks are solved using various methods. Some of them are shared among all tasks, such as representing words with fixed-size vectors or modeling the sequence of words in a sentence to obtain a representation for the target task. These methods employ algorithms from probabilistic modeling, information theory, optimization theory, and dynamic programming. Thus, NLP is a subject area which encompasses many tasks, methods, and algorithms.

In this work, we focus on named entity recognition (NER) which is one of the most utilized in NLP tasks. Its popularity can be demonstrated with a simple scenario. For example, consider a researcher who wishes to explore a news archive. Normally, she would begin by browsing the subject index to find the news items of interest to her. A typical subject index is often divided into a limited number of sections such as foreign relations, internal affairs, economics, petty crimes, sports related articles and opinion articles. For some researchers, this level of detail is adequate. However, if the researcher needs a more detailed analysis of articles on health issues, she may very well be interested in being able to investigate the person, organization or location names mentioned in these articles.

The named entity recognition (NER) task was introduced to provide a solution to the problem of exploring vast text databases or detecting mentions to entities in news outlets. Figure 1.1 shows the output of a NER system for the title and first few sentences of an opinion article[1] in The Atlantic. It identifies several organizations,

[1]The article was accessed on 09/04/2020 at https://www.theatlantic.com/ideas/archive/2020/03/interview-francis-collins-nih/608221/.

persona, and location. Suppose that the researcher is a historian from the future and wants to know the frequency of the articles about the COVID-19 pandemic that refer to the National Institute of Health in USA. Using NER systems, it is possible to build a searchable database of news articles that are related to a common named entity. The researcher would then be free to select 'NIH' or 'National Institute of Health' entities along with the 'coronavirus' keyword to specify the desired articles. If the researcher is restricted to browse the articles according the news sections, several articles of interest may be missed. For example, the article is in the *Opinion* section and would be missed by a person browsing the health section even though it is very relevant.

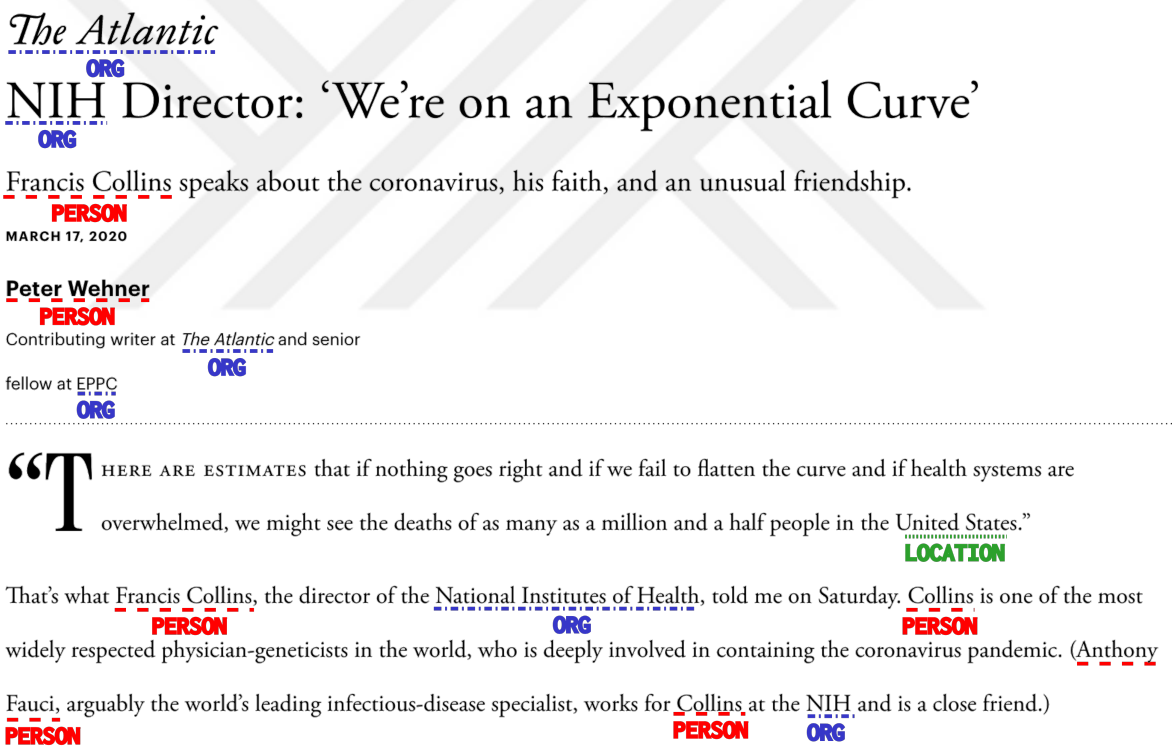



Figure 1.1. An excerpt from an opinion article published at *The Atlantic*. The NER system used while creating this illustration was targeting organization, person and location names, each marked with 'ORG', 'PERSON', and 'LOCATION' labels.

The NER task was first introduced in the Sixth Message Understanding Conference (MUC-6) [1] as a short-term subtask. At that time, it was thought that a practical system could be developed in a relatively short time, which could also serve as a domain-independent tool for other information extraction tasks. To accomplish

the NER task, portions of text were selected that refer to rigid designators. A rigid designator is some text that refers to the same entity in all possible contexts which they exist, and does not refer to anything else in contexts which that entity does not exist [2]. In practice, a rigid designator is almost always a proper name, except materials and objects that are found abundantly in nature, like water, air, apple, etc. For the purposes of this work, we limit the rigid designators to the most prominent ones: person names, geographical locations, and organization names, which were defined in the NER task in MUC-6. These designators are referred to as 'entity types' in our work.

The early work on NER often used word lists (known as gazetteers) that contain all known named entities for each entity type to detect named entities. However, this approach has some shortcomings. First of all, a single entity might be referred to in a variety of manners: the terms "JFK", "Kennedy" or "John F. Kennedy" may all refer to the same entity (the 35th president of USA) in relevant contexts. Secondly, the phrases "JFK" or "John F. Kennedy" might refer to the airport in New York City. This complex task has been addressed by numerous studies that will be detailed in Section 2.1. Current approaches employ sequence-based neural network approaches and incorporate word embeddings formed using unlabeled text from various domains [3–6]. However, these approaches are not well studied for morphologically rich languages (MRLs). Morphologically rich languages retain syntactical information in the morphology of the surface form of words. Such information is conveyed with the syntax of other types of languages. For example, in Turkish, a morphologically rich language, the word "İstanbul'daydı" means 'he/she was in Istanbul'. This single word embodies the tense and the locative case, which in this case also forms a complete sentence. This makes morphological understanding more important for MRLs in comparison to languages with simpler morphological mechanisms.

The hypothesis of this work is that morphological tags capture syntactic and semantic information and thus can be influential in improving the NER performance. The motivation behind this can be explained using the Turkish word "İstanbul'daydı". The morphological analysis of this word is given below. We follow the notation that

originates from the widely accepted representation scheme for Turkish [7].

```
İstanbul+Noun+Prop+A3sg+Pnon+Loc^DB+Verb+Zero+Past+A3sg
```

The analysis denotes that the root word 'İstanbul' is a proper noun in locative case and it is not marked with a possessive marker (see Section 3.2.1.1). The final word is a verb in past tense of 3rd person singular agreement. Looking at this analysis, one can suggest that the locative case tag and the fact that it is not possessed by anyone or anything might be a good indicator of it being a named entity.

If our hypothesis holds true, models that exploit the morphological information more effectively are expected to achieve higher NER task performance for morphologically rich languages. For this purpose we propose neural sequence tagger based NER models that exploit morphological analyses. In Chapter 3 we present a model that makes use of an external morphological disambiguator. Chapter 4 presents an approach that eliminates this dependency with a NER tagger that jointly learns to disambiguate morphological analyses. One may consider that earlier work that utilizes character based embeddings in word representations [3] and entities tagged at character level [8] already capture morphological information. Moreover, morphological tags have already been utilized for the NER task [9,10]. Our models process the morphological analysis in a number of different ways which can be applied to many morphologically rich languages, such as forming morphological tag sequences or treating the analysis as a character string. It is also the first to propose an embedding based framework for representing the morphological analysis in the context of NER. In Chapter 3, we show that when morphology is represented with embeddings the performance of the sequence based NER models surpass the state-of-the-art results. Building on this model, we introduce another model which eliminates the need for an external morphological disambiguator in Chapter 4. This is significant since external morphological disambiguators for many languages are not available, which limits the applicability of the former model. We remove this obstacle by incorporating a disambiguator in the second model. The disambiguator is jointly learned along with the tagger by using either

a single dataset that includes both morphological and named entity labels or several datasets consisting of some with morphological labels and others with named entity labels.

Our work also addresses the need to explain the predictions of a machine learning model. A user of our model might need an explanation for two reasons: i) to inspect whether the trained model is paying attention to parts of the input that make sense ii) to inspect the contributing factors given a specific input. In Chapter 5, we introduce a framework for explaining the predictions of sequence-based NLP tasks. This framework can be applied to any sequence-based model targeted at any NLP task given the probabilities for each possible prediction. Using this framework, we show how to provide an explanation for the named entity labels predicted by our models. Explanations are provided in terms of the significance attributed to each morphological feature associated with the tokens comprised in named entities. Figure 1.2 shows an explanation for a Finnish sentence.

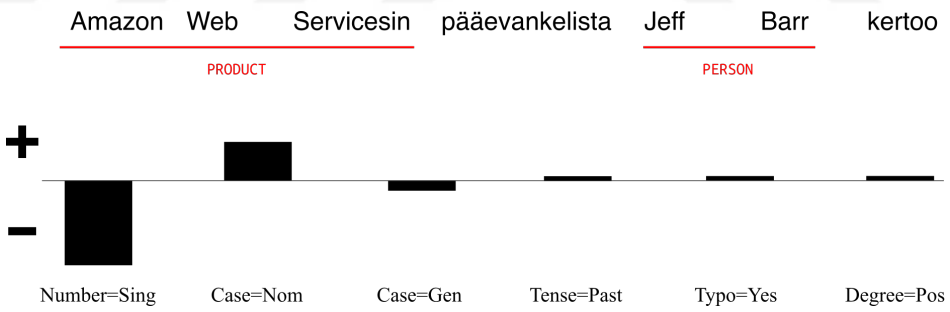



Figure 1.2. An explanation regarding the prediction of `PRODUCT` for the first three words of a Finnish sentence. Each bar is associated with a specific morphological feature. The height of a bar indicates the importance factor while the direction indicates the effect to be positive or negative.

## 1.1. Contributions

The main contributions of this thesis can be summarized as follows:

**The significance of morphological information in NER.** We showed that using morphological information to predict named entity tags improves the perfor-

mance. The experiments that are reported in Table 3.3 show that using embeddings based on morphological features yields higher performances for five languages (Turkish, Czech, Hungarian, Finnish, and Spanish) compared to embeddings based only on character sequences of the surface forms of the input words. Relying only on pre-trained word embeddings, the F1-measure for Turkish is 82.25%. The inclusion of character-based embeddings increases the F1-measure to 86.70%. When morphological embeddings are introduced, the performance reaches 88.12%. Finally, when both morphological and character-based embeddings are utilized the performance reaches 91.04%. When the model size is increased by modifying the cell size of bidirectional Long Short-Term Memory (Bi-LSTM) modules, our model achieves an F1-measure of 92.93% which became the state-of-the-art for Turkish. This model also surpasses the state-of-the-art for other four languages as reported in Table 3.4. Parts of this work has been presented in the 26th Signal Processing and Communications Applications Conference [11] and published in *Natural Language Engineering* [12].

**Joint morphological disambiguator and named entity tagger.** We showed the feasibility of building a named entity tagger, which itself disambiguates the morphological tags of words. This tagger exhibits comparable performance to those that utilize the disambiguated analyses obtained from external morphological disambiguators. Chapter 3 describes the three models we developed for this purpose: i) `EXT_M_FEAT`: a model that employs an external morphological disambiguator, ii) `NER`: the same architecture that does not use morphological features, and iii) `J_MULTI`: a model that jointly learns both how to label named entities as well as how to disambiguate morphological analyses. A comparison of `NER` (81.07%) with `J_MULTI` (83.21%) suggests that the morphological information improves the performance (Chapter 3). However, statistical tests to reject the hypothesis that the performances of `J_MULTI` and `EXT_M_FEAT` are on the same level, could not confidently reject the hypothesis. Although this does not prove the alternative hypothesis that one of them is better than the other, it does support that `J_MULTI` is on par with `EXT_M_FEAT`. This is significant as `J_MULTI` does not require external disambiguators, which are not readily available for many languages. This work has been presented at COLING 2018 [13].

**Explanation framework for any sequence-based NLP task.** A model-agnostic explanation method has been proposed for sequence-based NLP tasks. For this purpose, the concept of a region that can be adapted to any NLP task has been formalized. Given the rules for identifying a region for an NLP task, the method produces a vector consisting of importance scores corresponding to each feature of the model. This method is applied to explain the NER tagger introduced in Chapter 4. A qualitative and quantitative analysis is performed on the resulting explanations. The readability and plausibility of the explanations were found to be consistent with linguistic knowledge. This work has been published in PLOS ONE [14].

## 1.2. Thesis outline

This thesis is outlined as follows. The relevant work on named entity recognition, morphological analysis, and interpretability of machine learning models are provided in Chapter 2. Chapter 3 introduces a NER tagger that utilizes morphological analyses from an external morphological disambiguator. The NER tagger in Chapter 4 jointly learns to disambiguate morphological analyses. In Chapter 5, we introduce an explanation method for sequence-based NLP tasks that utilizes the NER tagger in Chapter 4 as a use case. Chapter 6 lays out difficulties that restricted us, ideas that would lead to future work, and, certain restrictions or difficulties encountered when forming explanations due to the nature of the language input. Finally, Chapter 7 gives a discussion of the outcome of the work presented in this thesis to aid future research.

# 2. RELATED WORK

## 2.1. Named entity recognition

NER is closely related to complex natural language understanding tasks such as relation extraction [15], knowledge base population [16], and question answering [17–19].

Early studies proposed compiling lists of people, place and organization names and exploiting them to decide whether there exists a named entity using hand crafted rules [20, 21]. Traditional approaches to NER typically use several hand-crafted features such as capitalization, word length, gazetteer based features, and syntactic features (part-of-speech tags, chunk tags, etc.). A wide range of machine learning-based methods have also been proposed to address the named entity recognition task. Some of the well known approaches are conditional random fields (CRF) [22, 23], maximum entropy [24], bootstrapping [25, 26], latent semantic association [27], and decision trees [28]. These techniques are generally used to create classification models which act on every token to decide whether there is an entity on that position of the text or not.

Recently, deep learning models have been instrumental in deciding how the parts of the input should be composed to form the most beneficial features leading to state of the art results [29]. One of the key issues is the determination of how to represent the words. This is due to the symbolic nature of words. These methods rely on simple tokenization by white space and employ distributional hypothesis [30]. The research in this direction led to the use of fixed length vectors in a dense space that improved the overall performance of many tasks, such as sentiment analysis [31], syntactic parsing [32], language modeling [33], part-of-speech tagging, and NER [29]. These word representations or embeddings are automatically learned either during or before the training phase using methods such as Word2Vec [34] GloVe [35]. The incorporation of morphology into this type of word embeddings was proposed for language model-

ing [36–40], and for morphological tagging and segmentation [41,42].

Built upon these findings, several approaches have been proposed which treat the NER task as a sequence labeling problem [3–6]. These studies employ Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU) components to capture the syntactic and semantic relations between the units that make up a natural language sentence. In Huang *et al.* [4], a Bi-LSTM network with a CRF layer on label scores is proposed. A special fixed-size representation is prepared for each word. The first component of this representation is the spelling features extracted from the surface forms such as whether all letters are lowercase or whether the first letter is uppercase. The second component is composed of a feature for each word bi-gram and tri-gram that exists in the context. The resulting representation is then fed to a word level LSTM. This network's NER performance is reported to be comparable to the performance of the state of the art studies. Another similar approach processes characters in each surface form in the sentence with a convolutional neural network (CNN) [5]. This helps the network to automatically represent the features that are extracted according to rules that are carefully designed in previous work. The word representations formed with this additional component is then fed to a word level LSTM resulting in state of the art performance. Two different approaches use LSTMs [3] and GRUs [6] instead of CNNs at the character and word levels otherwise similar to other work [5]. These two studies report results similar to each other and improve the state of the art results. In these studies, the morphological information present in the surface form of the word is handled only through the use of character based embeddings. Although this is not a limiting factor for languages which are not morphologically rich, in Chapter 3, we show that employing morphologically disambiguated tags when representing words in a neural architecture improves the NER performance.

There has been other approaches to the NER task for morphologically rich languages [9,10,43–45]. A study which can be considered as one of the first attempts in tackling NER for morphologically rich languages uses a hidden Markov model and takes the morphological tag sequence as input along with others like the surface form, capitalization features and similar features [9]. In a study which basically depends on

handcrafted features given to a CRF-based sequence tagger system, the morphology of the word was captured using the first and last three characters of the word as a feature resulting in an improvement in the NER tagging performance for Bengali [45]. In another study [10], a similar approach is taken with features generated using the output of an external morphological disambiguator and also shown to improve the performance. Another study [44] uses the same method but with a different approach for extracting morphological information, where they show an improvement over the previous state of the art results of [10]. The first study focusing on morphologically rich languages to employ neural networks [43] contains a regularized averaged perceptron [46] and relies on handcrafted rules along with pretrained word embeddings. However, they refrain from using output from external morphological disambiguators and only rely on the first and last few characters of a word as features. In our proposed NER tagger described in Chapter 3, however, the disambiguated morphological analysis is used to generate a fixed length vector in a number of different ways (Section 3.3.2) which is called a morphological embedding. This morphological embedding is composed with pretrained word embeddings and character based embeddings to obtain a word representation for each word and employed in a setting similar to previous work [5]. Our second proposed NER tagger described in Chapter 4 differs from these studies as it does not rely on handcrafted features. We represent words as fixed length vectors, employ morphological information to disambiguate the correct morphological analysis, and then combine them in such a way to obtain a context vector to label with NER tags.

## 2.2. Morphological analysis and disambiguation

In a recent study on morphological disambiguation [47], the authors propose a two-layer network for prediction. In the first layer, they process the candidate morphological analyses along with the correctly predicted analyses of previous words and obtain a vector to be processed in the second layer. The second layer takes all vectors propagated from the previous words and computes a `softmax` function over positive and negative classes. They predict the correct morphological analysis starting from

the first word and use this prediction in the next word positions. The model is evaluated on a dataset manually labeled by the authors and considered as the state of the art for Turkish and competitive for French and German. Our proposed model for morphological disambiguation relies on scoring the candidate morphological analyses to predict the correct one for a word in a sentence. We borrow this idea from [41]. In their study, they feed the word representations to a Bi-LSTM and obtain context embeddings for each position. Using these embeddings, they score each morphological analysis by calculating a similarity function reaching the state of the art or competitive results for Turkish, Russian and Arabic.

Most of the work in morphological disambiguation or tagging strictly depend on their chosen specific output format for morphological analysis. This is due to the fragmented nature of computational approaches to morphological analysis for every language in the literature. However, we argue that our approach is immune to this problem as all of these output formats can be treated as a sequence. An example from Finnish is '`raha+[POS=NOUN]+[NUM=SG]+[CASE=ADE]`' [48], another from Turkish is '`Ankara+Noun+Prop+A3sg+Pnon+Loc`' [7], and one for Hungarian is '`hír+NOUN+Case=Nom+Number=Plur`' [49]. All of these can be split by the '+' symbol and transformed into a root and tag sequence. Moreover, there is an attempt in the area to unify the morphological annotation along with syntax annotation across many languages which will contribute more towards a solution [50].

### 2.3. Joint modeling

Many models targeting NLP tasks are designed to work independently although they usually employ linguistic information related with other tasks. Given that there are state of the art models which are similar in the sense that they all employ a sentence level Bi-LSTM, it is reasonable to hypothesize that jointly learning several tasks will improve the performance as shown in the literature [51, 52]. In a recent study, it has been suggested that using separate layers for separate tasks is better rather than using the same (or usually top) layer for all the tasks [53].

## 2.4. Interpretability

There are several approaches to explaining the results of machine learning models. Some machine learning models are self-explanatory, such as decision trees, rule-based systems, and linear models. For example, the output of a decision tree model is a sequence of answers to yes/no questions, which can be considered as explanations. For other models, special mechanisms should be designed to provide explanations. Explanation models aim to provide two types of explanations: i) model (or global) explanations, and ii) outcome (or local) explanations. Local explanations focus on the outcome resulting from specific input samples, whereas model explanations reveal information about the machine learning model in question. Explanation models further differ in their explanation methods, the types of machine learning models that can be explained, and the type of data that can be explained [54]. According to the classification in [54], the method proposed in this work is a model-agnostic *features importance* explainer, since it aims to reveal the importance of each feature given an input sample.

The method proposed in this work (EXSEQREG) is inspired by the LIME approach [55] which explains the predictions of any model. It achieves this by perturbing the input to assess how the predictions change. LIME uses a binary vector to indicate whether a feature is perturbed, as described in Section 5.2. The binary vector ($z$) indicates the presence or absence of a feature. One shortcoming of this representation is that it does not convey whether a feature that is absent in the perturbed version is due to removal, since it may have been absent in the original input. In other words, a zero value may indicate two states: the feature does not exist at all or it is perturbed. Since this distinction may be significant in an explanation, we modified this scheme to remedy this. We follow an encoding scheme where we mark the features that are present but removed with minus one, the features that are present and not removed with one, and the features that are not present at all with zero. Furthermore, we focus on the probability differences induced by perturbations as opposed to the exact probabilities that are utilized by LIME.

A recent approach called LORE [56] learns a decision tree using a local neighborhood of the input sample. The method then utilizes this decision tree to build an explanation of the outcome by providing a decision rule to explain the reasons for the decision and a set of counterfactual rules to provide insights about the impacts of the changes in the features.

The method proposed in our work is part of a general class of methods called additive feature attribution methods [57]. Methods from this family include DeepLIFT [58], layer-wise relevance propagation method [59], LIME [55], and methods based on classic Shapley value estimation [60–62]. DeepLIFT aims to model the impact of altering the values assigned to specific input parts. Layer-wise relevance propagation works similar to DeepLIFT, however, in this case, the altered values are always set to zero. Shapley value estimation depends on the average of prediction differences when the model is trained repeatedly using training sets perturbed by removing a single feature $i$ from a subset $S$ of all unique features. Sampling methods for efficiently computing Shapley values are also offered [62]. All these methods, including LIME, depend on solving linear models of binary variables similar to Equation 5.1.

An explanation method for NER based on LIME has been proposed by [63]. The method treats input sentences as word sequences and ignores fine-grained features such as part-of-speech tags which are often attached to words as part of the input. The resulting explanation is a vector of real numbers that indicate the impact of each word. The method is restricted to models that are limited to token-level named entity tag prediction. However, every token is dependent on each other in the named entity recognition task. Many models exploit this dependency and combine token-level named entity tag prediction probabilities to have a single named entity prediction probability for the entire token sequence of the entity. Contrary to this method, in this work, we aim to handle this dependency issue by proposing a special transformation procedure, which is detailed in Section 5.3.4 section for named entity recognition.

LIME defines a text classification problem conditioned on features that correspond to the frequency of unique words in the input sentence. To obtain the explana-

tion vector for a given prediction, the input is perturbed by selecting a random set of words and eliminating all instances from the input sentence, thus removing the bag-of-words frequency values for the corresponding words from the input. This causes problems while explaining models that employ sequence processing constructs like recurrent neural networks (RNN) because a bag-of-words feature is devoid of information about the positions of the words within a sentence. This makes it difficult to relate a specific feature to a certain portion of the input sentence. Instead, selecting random subsequences (or substrings if we ignore tokenization) from the input sentence, designating these as distinct features, and removing these new features would both perturb the original sentence and enable specifying the specific position of the perturbation. These position-aware features are used in another extension of LIME to explain the prediction of such models [64]. In this work, however, we are only interested in the impact of features in a specific region in the sentence, so it is not required to have position-aware features.

The interpretation of machine learning models for NLP became significant subsequent to the success achieved by neural models. Although they achieve state-of-the-art results for many tasks, their black box nature leaves scientist curious about whether these models learn relevant aspects. For non-neural models, the approach was to provide mechanisms for explanations of features and their importance. However, the complexity of neural models has rendered explanations for models or specific outcomes very difficult. One approach is to use auxiliary diagnostic models to assess the amount of linguistic knowledge that is contained in a given neural representation [65–69].

Another prominent approach is to exploit attention mechanisms in the models [70, 71] to explain specific outcomes by attaching importance values to certain input features, like n-grams, words, or characters that make up the surface forms. Most of these methods modify the input samples so that they are reflected as changes in the output or the inner variables of the models. Other works exploit specially created datasets to assess the performance of an NLP task. For example, a custom dataset derived from a corpus of tasks related to the theory of mind was used to explore the capacity of a question answering model to understand the first and second-order

beliefs and reason about them [72]. Custom datasets are also used when trying to test whether the semantic properties are contained in word representations by using a special auxiliary diagnostic task that aims to predict whether the word embedding contains a semantic property or not [73].

Finally, approaches have been proposed to explain machine learning models by introducing latent variables to models [74] or that produce inherently interpretable output such as via word alignment information in machine translation [75].

## 2.5. IOB and IOBES tagging schema for NER

Named entities are labeled with types, such as 'PER' (person) and 'LOC' (location). The IOB scheme uses particular prefixes within a chunk to indicate whether it is inside (I), outside (O), and beginning (B) [76]. The IOBES scheme further refines this scheme to indicate the ending token and single tokens with the 'E' and 'S' prefixes respectively.

The labels are denoted as the position prefix followed by '-' and the type of the entity. Thus, a named entity of 'LOC' type consisting of a single token would be labeled as 'S-LOC'. A named entity of 'LOC' type that consists of three tokens would be labeled as 'B-LOC', 'I-LOC', and 'E-LOC' respectively. All tokens that are not a part of any named entitiy are labeled as 'O'.

# 3. MORPHOLOGICAL INFORMATION FOR NAMED ENTITY RECOGNITION

## 3.1. Introduction

In this part of the thesis, we introduce a neural named entity recognition model. This model treats the morphological analysis in a number of different ways which can be applied to many morphologically rich languages. This is the first work that proposes an embedding based framework for representing the morphological analysis in the context of NER.

The main contribution of this work is a state-of-the-art system for named entity recognition in morphologically rich languages. We also provide evidence that shows augmenting word representations with morphological embeddings improves NER performance.

This chapter is organized as follows: Section 3.2.1 provides information about morphologically rich languages in general and also some specific details about Turkish, Finnish, Czech, Hungarian, and Spanish. Section 3.2.2 covers Bi-LSTM networks utilized in the proposed model. In Section 3.3, the proposed model is defined. Section 3.4 presents the experiments in these five languages to evaluate our model's NER performance. The results of these experiments and a comparison of the proposed model's state-of-the-art results with the previous work are also given. Section 3.5 makes concluding remarks.

## 3.2. Background

### 3.2.1. Morphologically Rich Languages

The languages in which some of the syntactical functions of words are expressed by morphological phenomena within the surface forms of the words are called morphologically rich languages. In these languages, the number of words that can be generated from a single root word is very high in general. A significant number of inflections and derivations are possible for most root nouns and verbs. In practice, however, this potential is not fully realized where a few of the affixes are attached to a stem in succession to form new words. Regardless, the number of words that can be obtained from the root words is very large. This expressiveness gives rise to complications in applications, such as data sparseness due to words with many alternative affixes.

The following sections describe the basic characteristics of the morphologically rich languages Turkish, Czech, Hungarian, Finnish and Spanish.

3.2.1.1. Turkish. Turkish is an agglutinative language, which expresses most syntactic information through the morphology of the surface form of the words. Thus, studies in Turkish NLP have mainly focused on the morphological analysis of words. To this end, a finite state transducer based on a two level formalism [77] was introduced [7] to capture the rules of Turkish morphology [78,79]. The notation introduced in this work is considered the standard for morphological analysis in Turkish. The morphological analysis of the word "İstanbul'daydı" ('he/she was in İstanbul') is:

```
İstanbul+Noun+Prop+A3sg+Pnon+Loc^DB+Verb+Zero+Past+A3sg
```

where `Prop` indicates a proper noun, `A3sg` denotes the third singular person agreement, and `Pnon` signifies that no possessive agreement is active. `DB` (derivational boundary) indicates a transition of the part-of-speech usually induced by a derivational suffix. It

also marks the beginning of a new sequence of inflectional morphemes called inflectional group (IG) [80]. In this example, the derivation is triggered by the '-ydı' suffix which is decoded with the `Past` (past tense) tag.

3.2.1.2. Finnish. Finnish is an agglutinative language which exhibits vowel harmony and consonant gradation. Finnish uses derivational suffixes to a great extent. In our work, the morphological tagging of Finnish text is done by a toolkit called FinnPos [81], which is an averaged structured perceptron classifier. FinnPos relies on Omorfi [82] for morphological labels and lemmatization. In this work, the tool named `ftb-label` from the FinnPos package was used to obtain the disambiguated analysis as the morphological information associated with the word.

An example output of the disambiguator for the word 'Tampereella' meaning "in Tampere", where Tampere is a city in southern Finland is:

```
tampere+[POS=NOUN]+[PROPER=PROPER]+[NUM=SG]+[CASE=ADE]
```

which tags the word as a singular proper noun in adessive case.

3.2.1.3. Czech. Czech is a language with free word order known for its rich morphological properties. In our work, we obtain the morphological tags for Czech from a subset of the Prague Dependency Treebank 2.0 [83, 84]. These tags consist of 15 character strings where each position encodes a different morphological aspect. The tags in this treebank were labeled by seven annotators in two phases. In the first phase, annotators were given the output of a morphological tagger and were requested to select the best option. In the second phase, the discrepancies in annotator responses were resolved by another person [85]. For example, the word 'dlaně' ('palms') is decoded as:

```
dlaň+NNFP2-----A----
```

which tags the word as a common noun (NN), feminine (F), plural (P), genitive (2), and not-negated/affirmative (A). The tag can be decoded by a simple lookup in the tables provided in the morphological annotation manual for the Prague Dependency Treebank 2.0 [86].

3.2.1.4. Hungarian. Hungarian is also a morphologically rich language with free word order. In this work, the morphological features produced by the magyarlanc tool are associated with words [87]. There are three main morphological coding schemes for Hungarian: Humor [88], MSD [89], and KR [90]. We use the harmonized version of these schemes offered by the magyarlanc toolkit [91].

The morphological features for 'nyelvészek' ('linguists') are:

```
nyelvész+Case=Nom+Number=Plur
```

which indicates the nominal case and plurality.

3.2.1.5. Spanish. Spanish is a Romance language which is not generally considered as a morphologically rich language, but a fusional language. We examine it along with morphologically rich languages because it differs from many languages as the number of conjugated forms per verb can be as high as 47. A simplified version of the part of speech (POS) tags from the AnCora treebank[2] was used to obtain the morphological tags of the words. These POS tags do not specify the morphological features separately (as in the other four languages), instead they carry morphological information attached to the word. For instance, the Spanish word 'visitada' is labeled as VMP indicating the past participle form of the verb. Here, V indicates a verb, M states that the verb is principal, and P indicates a participle verb.

For sentences within the corpus, the supplied POS tags are used, while for sen-

[2]See http://clic.ub.edu/corpus/en/ancora and https://web.archive.org/web/20160325024315/http://nlp.lsi.upc.edu/freeling/doc/tagsets/tagset-es.html

tences outside the corpus, the Stanford POS tagger is employed.

### 3.2.2. Bidirectional LSTMs

Bi-LSTM is a type of sequential neural network which is composed of two long short-term memory (LSTM) modules, in which the first one reads the input sequence in forward order and the second one reads it in reverse order. The backwards component is important as it captures information about subsequent words, which can be highly significant in NLP tasks.

LSTM models were introduced to solve the vanishing gradients problem of recurrent neural networks (RNN) [92]. The sequential formulation of RNN is defined in terms of the following functions given a sequence of input vectors $x_t$ of size $d$:

$$\begin{aligned} h_t &= \tanh(Ux_t + Wh_{t-1}) \\ o_t &= \text{softmax}(Vh_t) \end{aligned}$$

where $h_t$ is the hidden state corresponding to item $t$ of size $p$, $o_t$ is the output vector corresponding to item $t$ of size $p$, $U$ is a $p \times d$ matrix, and $W, V$ are matrices of size $p \times p$.

LSTM differs from RNN in how it computes the output and hidden vectors. The following equations define an LSTM architecture [92]:

$$\begin{aligned} i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{t-1}) \\ f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{t-1}) \\ o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{t-1}) \\ \tilde{c}_t &= \tanh(W^{(c)}x_t + U^{(c)}h_{t-1}) \\ c_t &= f_t \circ c_{t-1} + i_t \circ \tilde{c}_t \\ h_t &= o_t \circ \tanh(c_t) \end{aligned} \tag{3.1}$$

where $\sigma$ is the sigmoid function and $\circ$ is the element-wise multiplication. In this architecture, variables $i_t$, $f_t$, and $o_t$ represent the will of the LSTM to *memorize* the cell's new value $\tilde{c}_t$, *forget* the contents of the previous cell's value $c_{t-1}$ and *display* the current cell's value $c_t$, respectively. The information is carried by $c_{t-1}$ and $h_{t-1}$ from the previous item. In this architecture, the hidden state $c_t$ is calculated as a parametrized sum of the previous hidden state $c_{t-1}$ and $\tilde{c}_t$, which is a nonlinear function of the input $x_t$ and the previous output $h_{t-1}$. This eliminates the repetitive multiplication performed in the RNN architecture, thereby solving the vanishing gradient problem.

For a single LSTM cell, the output vector $h_t$ can be used for classification, regression or as input to upper layers of the neural network. In the case of Bi-LSTM, two LSTM cells, one for forwards and one for backwards, with their corresponding output vectors $\overrightarrow{h_t}$ and $\overleftarrow{h_t}$ are defined. The concatenation $h_t = [\overrightarrow{h_t}; \overleftarrow{h_t}]$ is the output of a Bi-LSTM component [93].

An LSTM cell is trained by learning the variables, the matrices $W^{(i)}$, $W^{(f)}$, $W^{(o)}$, $W^{(c)}$ with size $p \times d$, and the matrices $U^{(i)}$, $U^{(f)}$, $U^{(o)}$, $U^{(c)}$ with size $p \times p$. Learning is performed by backpropagation through time [94] with an update rule of choice. A simple choice such as vanilla stochastic gradient descent or more advanced gradient based algorithms like Adam [95] or RMSProp [96] can be used. The training of Bi-LSTM is the same with additional parameters for the extra LSTM.

## 3.3. Model

In this work, the named entity recognition problem is treated as a sequence tagging problem. The input is expected to be in sentence form and the entities referred in it must be labeled with the IOB scheme (Section 2.5). Figure 3.1 shows a labeled English sentence. The sample in the figure contains three named entities: 'UN' as an organization name, 'Rolf Ekéus' as a person name, and 'Baghdad' as a location name.

Given an input sentence of length $n$ and its corresponding labels, we define them as $X = (x_1, x_2, \ldots, x_n)$ and $Y = (y_1, y_2 \ldots, y_n)$. Each $x_i$ is a fixed length vector of

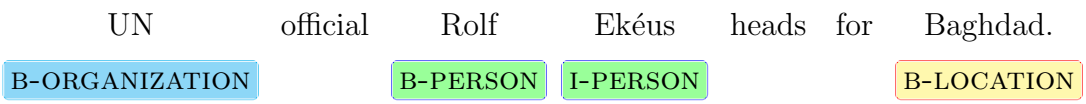



Figure 3.1. NER sample with IOB tagging scheme.

size $d$, consisting of embeddings that represent the $i$th word. Further details of word representations are provided in Section 3.3.1. The label variable $y_i$ is a vector of size $K$ such that $y_{ik} = 1$ if and only if the correct tag is the $k$th tag in our tag vocabulary of size $K$, otherwise $y_{ik} = 0$. Tag vocabulary when IOB scheme is used consists of 'B-tag' and 'I-tag' for each tag, and 'O'.

The words $x_i$ are fed to a Bi-LSTM which is composed of two LSTMs [92] treating the input forwards and backwards respectively as explained in Section 3.2.2. The forward and backward components' cell matrices $\overrightarrow{H}$ and $\overleftarrow{H}$ are both of size $n \times p$, where $p$ is the number of dimensions of one component of the Bi-LSTM. Figure 3.2 describes how the proposed model works with a Turkish sentence as an example. $\overrightarrow{H}_{i,j}$ denotes the value of the $j$th dimension of the $i$th output vector of the forward component which corresponds to the $i$th word in the sentence, whereas the corresponding value in the backward component is denoted by $\overleftarrow{H}_{n-i+1,j}$. The concatenation of rows $\overrightarrow{H}_i$ and $\overleftarrow{H}_{n-i+1}$ from $\overrightarrow{H}$ and $\overleftarrow{H}$, respectively, are fed to a fully connected hidden layer of $K$ output neurons for each of the $n$ input words. The output of this fully connected layer for the $i$th word is denoted with $\xi_i$.

A conditional random field (CRF) [97] based approach is followed to predict the entity tags. CRF based approaches model the dependencies between consecutive input units better than the approaches that only try to predict the correct tag based on $\xi_i$. At this point, it is possible to exponentiate the values of $\xi_i$ vectors and normalize them to obtain a vector which we utilize to estimate the probabilities of each tag: $\tilde{\xi}_{i,k} = \frac{\exp(\xi_{i,k})}{\sum_{\kappa=1}^{K} \exp(\xi_{i,\kappa})}$ and use $\tilde{\xi}_i$ in a cross-entropy loss function to optimize the parameters of the model: $s(x_i, y_i) = -\sum_{k=1}^{K} y_{i,k} \texttt{log}(\tilde{\xi}_{i,k})$ where $x_i$ is the $i$th word and $y_i$ encodes the correct tag for the $i$th word. However, this approach is weaker than using a loss function which also models the dependencies between the consecutive input units.

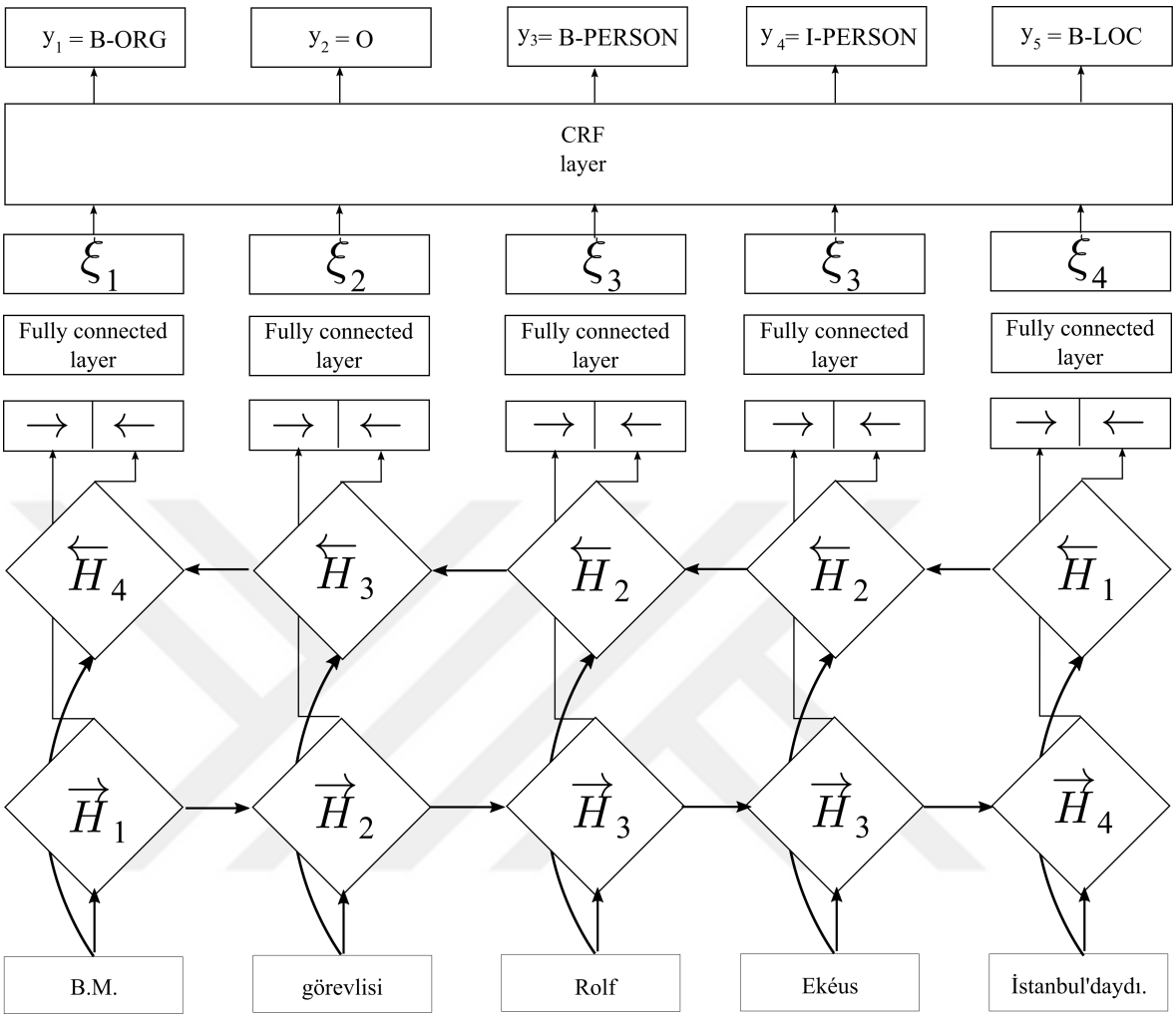



Figure 3.2. Processing a Turkish sentence with the proposed model which can be translated as 'UN officer Rolf Ekéus was in Istanbul'. Words are represented with a fixed length vector that combines the word, character, and morphological embeddings (see Section 3.3.2).

The advantage of the CRF model stems from the logical requirements of the IOB tagging scheme. For instance, the scheme allows tags that start with 'I-' only after a 'B-' or 'I-' tag of the same type. That is, 'I-PERSON' might only come after 'B-PERSON' or 'I-PERSON', and not after 'O' or 'B-LOCATION' or others. Moreover, the sequences in the corpus might indicate an ordering relation between two tag types. For instance, 'LOCATION' tags may tend to appear more frequently before 'PERSON' tags compared to the other way around. Since CRF models allocate a probability to every valid sequence, it is possible to determine which tag sequences do not adhere to these rules as well as assigning higher probability to sequences that are in line with the

ordering relations.

In order to implement a CRF model, the tag score vector $\xi_i$ at each position $i$ is treated as the observation score obtained from the fully connected layer and the following objective function for a sample sentence $X$ is optimized:

$$s(X, y) = \sum_i A_{y_i, y_{i+1}} + \sum_i \xi_{i, y_i} \tag{3.2}$$

where $A_{t,t'}$ represents the score of a transition from tag $t$ to tag $t'$. Then, the most probable tagging sequence $y^*$ is

$$y^* = \underset{y'}{\operatorname{argmax}}\, s(X, y').$$

### 3.3.1. Embeddings

It has been shown that modeling units of information in a natural language input as fixed length vectors, which are called embeddings, is more effective at encoding semantic properties of the words compared to using manually designed features [29, 98]. Our model in this chapter utilizes embeddings where the input words, $x_i$, are represented as fixed length vectors composed of three components: *word*, *character*, and *morphological*.

**Word embeddings.** For each unique word, a vector of length $d_w$ is defined. As mentioned above, word representations are connected to the final loss expression through $\xi_i$. This relation makes them parameters of the model. Thus, it is possible to optimize each of the $d_w$ dimensions for the target task for every unique word. However, these parameters are not learned from scratch during training of our model in this chapter. The word embeddings are initialized to vectors obtained through approaches like skipgram with negative sampling [34] and fastText [99]. If a corpus larger than the Wikipedia is available for a specific language, the skipgram algorithm is employed

to obtain the word embeddings using that corpus. If such a corpus is not available, we use the word embeddings that are pretrained using Wikipedia [99].

**Character embeddings.** In addition to the word embedding for a word, the covert relationships in the character sequence of the word is of value [5]. To capture these relationships, a separate Bi-LSTM component is used for this embedding type with a cell dimension of $d_c$. This Bi-LSTM component is fed with the characters of the surface form of the $i$th word. After all the characters are processed by the Bi-LSTM component, the last cell's output of the forward and backward LSTMs are concatenated to obtain the *character* embedding of the word of length $2d_c$ (see Figure 3.3).

**Morphological embeddings.** These embeddings are constructed similar to *character* embeddings. In this case, the tags of the morphological analysis are treated as a sequence and fed to a separate Bi-LSTM component for *morphological* embeddings, resulting in a vector of length $2d_m$. Our definition of a morphological analysis differs for each language. While all definitions are made up of morphological tags that are combined in some way, morphological tags do not share a common set across languages. In our work, we tried several alternative combination strategies for each language to obtain morphological tag sequences as described in Section 3.3.2.

When all of the components are active, the total size of a word representation is $d = d_w + 2d_m + 2d_c$. Figure 3.3 shows the representation for the Turkish word "evlerinde" ("in his/her houses"). This representation is then fed into the sentence level Bi-LSTM described in Section 3.3[3] .

[3]Experiments with three separate sentence level Bi-LSTMs, one for each of the components of our word representation were also done. However, initial experiments with this model did not give good enough results to proceed further. By doing this, it was thought that this might help in training the Bi-LSTMs so that they are better customized to their specific input embeddings. This is basically the same as the model described above. However in separate Bi-LSTM mode, although the outputs from each of the separate Bi-LSTMs are concatenated too, each separate Bi-LSTM is fed with only one component of the word representation. For example, morphological embeddings are fed as if only they are available, while word embeddings are fed into another Bi-LSTM and the character embeddings are fed into a third Bi-LSTM.

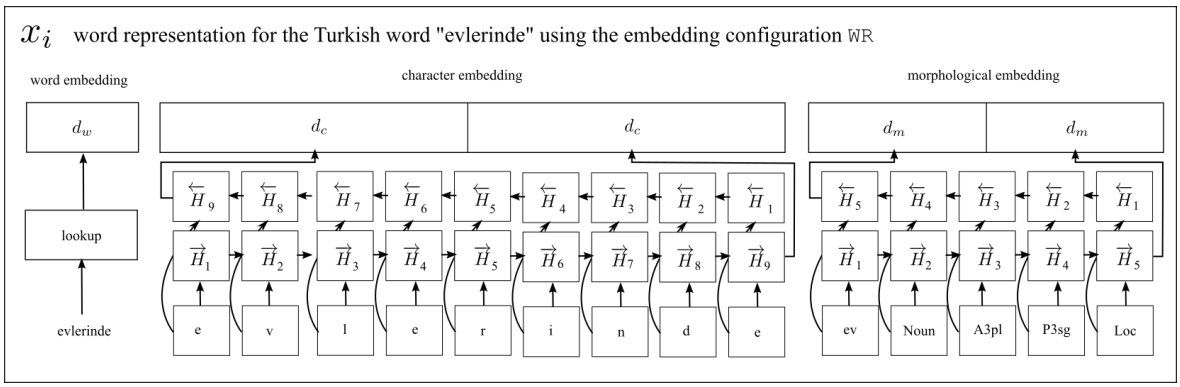



Figure 3.3. The representation of the Turkish word “evlerinde” according to the proposed model featuring all three components: word, character and morphological embeddings.

### 3.3.2. Morphological embedding configurations

In order to determine an effective configuration for extracting the syntactic and semantic information in the morphological analysis of a word, experiments with a total of four different combinations of morphological tags were performed. In morphologically rich languages, it is common for a word to have more than one possible morphological analysis. The correct analysis within a context is determined using a morphological disambiguator for that language. For example, the Turkish word ‘evlerinde’ has three different meanings depending on the context: ‘in their house’, ‘in their houses’ and ‘in his/her houses’. If the correct sense in a particular context is the last one, then the disambiguator will output the morphological analysis ‘ev+Noun+A3pl+P3sg+Loc’. Here ‘A3pl’ indicates 3$^{\text{rd}}$ person plural, ‘P3sg’ is the possessive marker for 3$^{\text{rd}}$ person singular, and ‘Loc’ is the locative case marker. After the correct morphological analysis for a word is determined, the embeddings are formed as explained below.

Table 3.1 shows an example for each morphological embedding configuration employed in this work for each language. The first one uses the root accompanied with all the morphological tags in the analysis. This embedding configuration is called `WITH ROOT (WR)`. This is the simplest embedding style that can be considered given the morphological analysis of words in any language. This is because most of the time the morphological tags output by morphological disambiguation tools can be treated as

sequences.

For Turkish, the morphological analysis of the word "evlerinde", which is "ev+Noun+A3pl+P3sg+Loc", is transformed into a list by splitting from the '+' symbols. For Czech, the output of the morphological disambiguator [100] are the root lemma and a string of 15 characters for tags. In the WITH ROOT (WR) configuration, this is transformed into a fixed length list by splitting with the '+' symbol. For instance, the analysis 'prezident+NNMS1-----A----' is converted into ('prezident', 'NNMS1-----A----'). A similar transformation is applied to Hungarian. The analysis of the word 'Magyar', "Magyar+PROPN+Case=Nom+Number=Sing", is converted into the list ('Magyar', 'PROPN', 'Case=Nom', 'Number=Sing'). The disambiguator that is used for Finnish outputs the tags separately so it is just transformed into a list. The disambiguated morphological analysis of the word 'Tampereella' is 'tampere+[POS=NOUN]+[PROPER=PROPER]+[NUM=SG]+[CASE=ADE]'. Then for the embedding configuration WR, it is transformed into ('tampere', '[POS=NOUN]', '[PROPER=PROPER]', '[NUM=SG]', '[CASE=ADE]'). One exception to this is Spanish in our choice of languages. For Spanish, a single tag is used to represent both the part of speech and the morphological properties that might be attached to the word. For instance, the Spanish word 'visitada' is only labeled with 'VMP' indicating the past participle form. So the result is just a single item list for Spanish.

The second embedding configuration follows from the WITH ROOT (WR) scheme. The root is omitted from the list and the resulting list is the embedding configuration WITHOUT ROOT (WOR). Obviously, this embedding configuration does not make sense for languages that do not carry the root lemma in their morphological analysis representations, such as Spanish.

The next configuration is specific to the Turkish language. The morphological notation of Turkish includes a 'DB' tag. This tag denotes a change in the part of speech during the derivational process, hence the name 'derivational boundary' DB. The parts that are separated by DB tags are called inflectional groups (IG) [80]. To examine the significance of these boundaries on the performance of the task, the tags between

the root and the last derivational boundary are removed. The intuition is that the information given by the features before the last derivational boundary may not be relevant at all or may be relatively less relevant. This is because the last derivational part yields the derived word whose lexical and syntactic properties may be slightly different from the intermediate parts. This configuration is named as `WITH ROOT AND AFTER LAST DB (WR_ADB)`. The following is an example for the word "İstanbul'daydı" ("he/she was in İstanbul"). The first line is the raw morphological analysis. The model uses the form in the second line.

```
1. İstanbul+Noun+Prop+A3sg+Pnon+Loc^DB+Verb+Zero+Past+A3sg
2. ('İstanbul', 'Verb', 'Zero', 'Past', 'A3sg')
```

Finally, the morphological analysis of a word is simply treated as a string, which is transformed to a list containing each of its characters. This embedding configuration is referred to as `CHAR`. Examples for all of these configurations can be observed in Table 3.1.

## 3.4. Experiments

To test the validity and performance of our proposed method, two main sets of experiments are conducted: i) experiments that compare the proposed approach with the state-of-the-art models, ii) experiments that aim to demonstrate the differing performance characteristics of different model configurations. In this section, before giving the results of the experiments, the training method and the datasets used in the experiments are explained.

### 3.4.1. Training

The parameters to be learned by the training algorithm are the parameters of the Bi-LSTM described in Section 3.3 (Figure 3.2), the parameters of the Bi-LSTMs for the *character* and *morphological* embeddings (Figure 3.3), and the word embeddings

for each unique word. After experiments with several different choices for the number of dimensions for these parameters, a choice of 100 for word embeddings, 200 for character embeddings, and 200 for morphological embeddings was observed to give the best results. However, it was not possible to use these dimension sizes in all of the experiments covering all of the languages and configurations due to time complexity.

In the first set of experiments that aim to compare the model configurations, the cell dimension of the sentence level LSTM, word, character and morphological embedding dimensions, and character and morphological LSTM cell dimensions were set as 10. For word embeddings, embeddings of size 10 were trained with an algorithm that takes sub-word information into account [99] using the corresponding language version of Wikipedia and were used as pretrained word embeddings. Higher dimension sizes that yield the best performances as stated above were used in the second set of experiments that compare the proposed model with the previous work.

Model training was done by calculating the gradients using the back propagation algorithm and updating the parameters with the stochastic gradient descent algorithm with a learning rate of 0.01. Gradient clipping was employed to handle gradients diverging from zero. Additionally, dropout was used on the inputs with probability 0.5. Each language was trained for 50 epochs.

### 3.4.2. Datasets

Five morphologically rich languages were selected to evaluate the proposed method: Turkish, Czech, Hungarian, Finnish, and Spanish. In this section, the dataset of each language is described separately as the morphological analysis format, the pretrained word embeddings, and the origin of the data differs for each.

**Turkish.** Our model is trained and evaluated using a corpus which was widely used in previous works on Turkish NER [9]. The training part of the corpus contains 14,481 person names, 9,411 location names, and 9,037 organization names while the test part contains 1,594 person names, 1,091 location names, and 863 organization

names. In addition to the named entity tags and the corresponding surface forms, the corpus also contains a single disambiguated morphological analysis for each input word.

Word embeddings[4] of Turkish words as vectors of length 100 were obtained using the skipgram algorithm [34] on a corpus of 951 million words, 2,045,040 of which are unique [101], This corpus consists of Turkish texts extracted from several national newspapers, news sites, and book transcripts. fastText algorithm was employed to obtain word embeddings of size 10 using the same corpus [99].

**Czech.** CNEC 2.0 corpus was used to test the performance of our model on the Czech language [83, 102]. Seven different named entity types are labeled in this corpus. The number of labels for each of these entity types for training, validation and test portions of the dataset is given in Table 3.2. For each word, the morphological analysis provided in the dataset was used. fastText algorithm was used to obtain pretrained word embeddings of size 10 and 100 for Czech using the Czech version of Wikipedia [99].

**Hungarian.** We used 'The Named Entity Corpus for Hungarian' which contains around 14,400 phrases tagged with entity labels [103]. The corpus is labeled with the standard named entity tags. The training part contains 795 person names, 1,056 location names, 8,458 organization names, and 1,327 miscellaneous names. The corpus originally contained only training and test parts, so validation and test sets were created by randomly selecting from the test part. The test set contains 100 person names, 125 location names, 1,055 organization names, and 160 miscellaneous names. In the validation set, there are 87 person names, 113 location names, 1,020 organization names, and 174 miscellaneous names. A statistical morphological analysis tool for Hungarian was used to process each word [87] and its output was used as the input for morphological embeddings. For word embeddings, fastText algorithm was used to obtain pretrained word embeddings of size 10 and 100 for Hungarian using the Hungarian version of Wikipedia [99].

[4]These word embeddings are available at https://github.com/onurgu/linguistic-features-in-turkish-word-representations/releases.

**Finnish.** A labeled corpus[5] which was compiled from news articles in an online Finnish technology news site was used. The articles were published between 2014 and 2015. Extracting the morphological tags was done by a Finnish morphological analysis tool called Omorfi. Morphological disambiguation was done by FinnPos while creating the training and test sets [81]. This corpus is labeled with five more named entity tags in addition to the standard set: '`DATE`' for depicting date references, '`EVENT`' for marking events, '`PRO`' for marking products, '`TIM`' for marking time expressions and '`TIT`' for titles. The number of labels for each entity type is shown in Table 3.2. fastText algorithm was used to obtain pretrained word embeddings of size 10 and 100 for Finnish using the Finnish version of Wikipedia [99].

**Spanish.** CoNLL 2002 Shared Task[6] publishes a corpus tagged with NER and POS labels which has clearly defined training, development and testing portions of the dataset. This dataset is widely used in NER related research for benchmarking. The POS tags were treated as the morphological analysis of the word as the POS tag contains the morphological information associated with the word if there is any. This corpus contains 6,278 person names, 6,981 location names, 10,490 organization names, and 2,957 names of miscellaneous types. fastText algorithm was used to obtain pretrained word embeddings of size 10 and 100 using the Spanish version of Wikipedia[7] [99].

### 3.4.3. Results

This section presents the results of experiments performed to measure the impact of using morphological information for the NER task along with character based embeddings. The experiments are conducted with the Turkish, Czech, Hungarian, Finnish and Spanish languages.

The experiments are performed with alternative embedding configurations, which

[5] We obtained the corpus from `https://github.com/mpsilfve/finer-data` for our experiments. At that time, the corpus was not part of a published article. It was later published [104].

[6] The data can be accessed at `http://www.lsi.upc.es/~nlp/tools/nerc/nerc.html`.

[7] `http://www.wikipedia.org`

are referred to with *Setup* followed by an integer as an identifier. The setups are: i) only pretrained word embeddings (Setup 1), ii) only word and character embeddings (Setup 2), iii) only word and a choice of one of the morphological embedding configurations (Setups 3-6), and iv) word, character, and a choice of one of the morphological embeddings (Setups 7-10).

Table 3.3 summarizes these results. A comparison of the basic model (Setup 1) with those that use morphological information (Setups 3-6) shows a performance increase when morphological information is used (`ME(CHAR)` and `ME(WOR)`).

In the case of Setup 4 an improvement is observed only for Turkish. Also, again for Turkish, using only the tags after the last derivational boundary (`ME(WR_ADB)`) is one of the most successful morphological configurations.

The performances of `ME(CHAR)` and `ME(WOR)` are comparable with the latter being slightly lower (except for Czech). The difference in the performance between Setups 5 and 6 may stem from the errors present in the morphological analyses. These errors are mostly due to unknown or misspelled words. In such cases, the analysis in the corpus usually defaults to the same nominal case. The higher performance of `ME(CHAR)` may be attributed to its ability to handle possibly faulty roots as this leads to morphological embeddings that capture more useful information in comparison to `ME(WOR)`.

Another advantage of `ME(CHAR)` embeddings is its ability to capture the relationship between roots with the same prefix. For example, in the Finnish corpus, the frequencies of words with the same prefix often differ significantly based on their roots, such as in the cases of 'allekirjoittaa' (sign) vs. 'allekirjoittaja' (signatory) and 'Tampere' (a city in southern Finland) vs. 'Tamperelainen' (of Tampere) where the occurrence of the former is much frequent than the latter. The `ME(CHAR)` scheme also benefits from the common parts in related morphological tags. For instance, in Turkish, the tags '`A3sg`' and '`A3pl`' denote, respectively, 3rd person singular and 3rd person plural, where the leading two characters 'A3' indicate 3rd person agreement. The model can capture this information when the tags are represented in terms of characters.

Therefore, ME(CHAR) is a better representation than either ME(WR) or ME(WOR).

Using only character embeddings in addition to the word embeddings (Setup 2) also improves the NER performance. Combining character embeddings with the ME(CHAR) and ME(WOR) models (Setups 9 and 10) outperforms all other setups (except Czech). For all the languages, the best performance is achieved with Setup 9 where both character embeddings and ME(CHAR) embeddings are employed[8]. Although adding CE to Setup 1 (word embeddings) causes a large improvement, adding CE to Setup 5 (word embeddings and character embeddings of morphological part) provides a relatively small increase in performance (Setup 9). This may be the result of CE and ME(CHAR) both taking part in representing morphological information of words.

The results of these experiments show that an increase in NER performance is observed for all languages when either of CE or ME is included in the word representation. The best performance is achieved when both CE and ME are included in the word representation for all languages.

Table 3.4 shows a comparison of the proposed approach in this chapter with the state-of-the-art results reported in literature. The CE+ME(CHAR) configuration (Setup 9) is used for comparison, since it yielded the best results. The models for each language were trained with higher number of parameters. The values for cell dimension of the sentence level LSTM, character and morphological LSTM cell dimensions, and character and morphological embedding dimensions were all set to 200. The word embeddings was set to 100. Other hyper-parameters and training related settings were unaltered. This work is the first one to report test results for all of Czech, Turkish, Hungarian, Finnish and Spanish. As such, comparisons are made using different studies for each language. For Turkish, a comparison with three different results are presented. The performance of Şeker and Eryiğit [44] shown in the table was obtained using gazetteers. When such resources are not used, the performance drops to 89.55%. Kuru *et al.* [8] do not employ any external data. Demir and Özgür [43] rely on hand-crafted

[8]The relatively lower values for Czech is due to the corpus that was used. This is also apparent in Table 3.4 where we compare our best results with the literature, i.e. the performance on Czech dataset of other work are also relatively lower compared to the performance on the Turkish dataset.

features, however exploit externally trained word embeddings. A new approach for Turkish was introduced in 2020 that exploits contextual word embeddings to tag NER labels [105]. This approach represents the current state-of-the-art for NER tagging with a F1-measure of 95.55%. The creators of the Finnish dataset were working on publishing a NER model during we make our own experiments. In the meantime, they have provided a performance metric using Stanford NER[9] . Recently, the authors published the results of their rule-based Finnish NER tagger FiNER [104]. The new results surpass our F1-value of 84.34 by 2.48 points. We believe this gap is the result of the patterns formed carefully during the design process of their rule-based tool. For Spanish, we chose to report the state-of-the-art result for Spanish NER task [3]. A noteworthy observation is the increase in NER performance for Spanish even though its morphological characteristic differs from the other languages.

## 3.5. Conclusions

In this chapter, a state-of-the-art system for named entity recognition in morphologically rich languages was introduced. Several ways for combining morphological tags to obtain fixed length vector embeddings that represent the morphological information were compared by their impact to the NER performance. It revealed that augmenting word representations with morphological embeddings improves NER performance, which is further improved when combined with character based word representations. Experiments with five languages, all morphologically rich languages except Spanish, were performed. The results obtained using this approach are the state-of-the-art for all of these languages. However, very recently, our Finnish result is surpassed by a rule based system [104]. Our Turkish result is also recently improved by an approach that utilizes contextual word embeddings [105]. An ablation study to examine the impact of using morphological information revealed that the improved performance was similar across these languages.

Although extensive experiments regarding the interaction of morphological anal-

[9] https://github.com/mpsilfve/finer-data/blob/79f652521873a87554e5c9ffce1416e8d22e1e60/documents/finer.tex

ysis based and surface form based representations were done, morphological tags and affixes which are subcomponents of these two representations were not investigated with much detail. In Chapter 5, we will introduce an explanation framework that is designed to assign importance values to parts of a given input to any NLP task. NER model in Chapter 4 is used as a use case.

Table 3.1. Embedding configurations for Turkish, Czech, Hungarian, Finnish and Spanish: WITH ROOT (WR), WITHOUT ROOT (WOR), WITH ROOT AND AFTER LAST DB (WR_ADB), CHAR.

| | | |
|---|---|---|
| TR | WR | (‘ev’, ‘Noun’, ‘A3pl’, ‘P3sg’, ‘Loc’) |
| | WOR | (‘Noun’, ‘A3pl’, ‘P3sg’, ‘Loc’) |
| | WR_ADB | (‘Istanbul’, ‘Verb’, ‘Zero’, ‘Past’, ‘A3sg’) |
| | CHAR | (‘e’, ‘v’, ‘+’, ‘N’, ‘o’, ‘u’, ‘n’, ‘+’, ‘A’, ‘3’, ‘p’, ‘l’, ‘+’, ‘P’, ‘3’, ‘s’, ‘g’, ‘+’, ‘L’, ‘o’, ‘c’) |
| CS | WR | (‘prezident’, ‘NNMS1-----A----’) |
| | WOR | (‘NNMS1-----A----’) |
| | CHAR | (‘p’, ‘r’, ‘e’, ‘z’, ‘i’, ‘d’, ‘e’, ‘n’, ‘t’, ‘+’, ‘N’, ‘N’, ‘M’, ‘S’, ‘1’, ‘-’, ‘-’, ‘-’, ‘-’, ‘-’, ‘A’, ‘-’, ‘-’, ‘-’, ‘-’) |
| HU | WR | (‘Magyar’, ‘PROPN’, ‘Case=Nom’, ‘Number=Sing’) |
| | WOR | (‘PROPN’, ‘Case=Nom’, ‘Number=Sing’) |
| | CHAR | (‘M’, ‘a’, ‘g’, ‘y’, ‘a’, ‘r’, ‘+’, ‘P’, ‘R’, ‘O’, ‘P’, ‘N’, ‘+’, ‘C’, ‘a’, ‘s’, ‘e’, ‘=’, ‘N’, ‘o’, ‘m’, ‘+’, ‘N’, ‘u’, ‘m’, ‘b’, ‘e’, ‘r’, ‘=’, ‘S’, ‘i’, ‘n’, ‘g’) |
| FI | WR | (‘tampere’, ‘[POS=NOUN]’, ‘[PROPER=PROPER]’, ‘[NUM=SG]’, ‘[CASE=ADE]’) |
| | WOR | (‘[POS=NOUN]’, ‘[PROPER=PROPER]’, ‘[NUM=SG]’, ‘[CASE=ADE]’) |
| | CHAR | (‘t’, ‘a’, ‘m’, ‘p’, ‘e’, ‘r’, ‘e’, ‘+’, ‘[’, ‘P’, ‘O’, ‘S’, ‘=’, ‘N’, ‘O’, ‘U’, ‘N’, ‘]’, ‘+’, ‘[’, ‘P’, ‘R’, ‘O’, ‘P’, ‘E’, ‘R’, ‘=’, ‘P’, ‘R’, ‘O’, ‘P’, ‘E’, ‘R’, ‘]’, ‘+’, ‘[’, ‘N’, ‘U’, ‘M’, ‘=’, ‘S’, ‘G’, ‘]’, ‘+’, ‘[’, ‘C’, ‘A’, ‘S’, ‘E’, ‘=’, ‘A’, ‘D’, ‘E’, ‘]’) |
| ES | WR | (‘VMP’) |
| | CHAR | (‘V’, ‘M’, ‘P’) |

Table 3.2. The number of labels for each entity type in Czech and Finnish datasets.

| Czech | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | Person | Geo. | Inst. | Media | Address | Time | Artificial | | |
| Training | 3,757 | 3,117 | 2,705 | 314 | 402 | 2,431 | 2,459 | | |
| Validation | 509 | 431 | 340 | 53 | 77 | 280 | 325 | | |
| Test | 480 | 378 | 324 | 48 | 55 | 368 | 382 | | |
| Finnish | | | | | | | | | |
| | Person | Location | Org. | Misc. | Date | Event | Product | Time | Title |
| Training | 2,229 | 2,040 | 9,098 | 907 | 956 | 93 | 4,462 | 4,958 | 631 |
| Test | 409 | 505 | 1,910 | 182 | 238 | 17 | 1,134 | 1,066 | 129 |

Table 3.3. The performance of the model using various embedding configurations for five languages.

| | **Setups** | | **F1-Measure** | | | | |
|---|---|---|---|---|---|---|---|
| **Setup** | **CE** | **ME** | **TR** | **CS** | **HU** | **FI** | **ES** |
| 1 | - | - | 82.25 | 67.56 | 94.02 | 70.56 | 80.38 |
| 2 | CE | - | 86.70 | 72.35 | 95.10 | 79.36 | 81.00 |
| 3 | - | ME(WR_ADB) | 87.99 | N/A | N/A | N/A | N/A |
| 4 | - | ME(WR) | 87.78 | 66.62 | 93.98 | 67.30 | N/A |
| 5 | - | ME(CHAR) | 88.12 | 72.66 | 95.11 | 75.89 | 82.19 |
| 6 | - | ME(WOR) | 87.78 | 67.85 | 95.14 | 75.34 | 81.44 |
| 7 | CE | ME(WR_ADB) | 87.69 | N/A | N/A | N/A | N/A |
| 8 | CE | ME(WR) | 87.09 | 69.10 | 92.67 | 72.17 | N/A |
| 9 | CE | ME(CHAR) | **91.04** | **73.61** | **95.60** | **81.37** | **82.94** |
| 10 | CE | ME(WOR) | 89.85 | 67.19 | 95.50 | 80.27 | 82.68 |

Table 3.4. Comparison of results with state-of-the-art NER results for each language.

| **Work** | **F1-Measure** | | | | |
|---|---|---|---|---|---|
| | **TR** | **CS** | **HU** | **FI** | **ES** |
| Kuru *et al.* [8] | 91.30 | 72.19 | - | - | - |
| Demir and Özgür [43] | 91.85 | 75.61 | - | - | - |
| Şeker and Eryiğit [44] | 91.94 | - | - | - | - |
| Varga *et al.* [106] | - | - | 94.77 | - | - |
| Lample *et al.* [3] | - | - | - | - | 85.75 |
| Strakova *et al.* [107] | - | 80.79 | - | - | - |
| (unpublished, uses Stanford NER) | - | - | - | 82.42 | - |
| The model in this chapter | **92.93** | **81.05** | **96.11** | **84.34** | **86.95** |
| BERTurk [105] | **95.55** | - | - | - | - |
| Ruokolainen *et al.* [104] | - | - | - | **86.82** | - |

# 4. NAMED ENTITY RECOGNITION AND MORPHOLOGICAL DISAMBIGUATION WITH A JOINT MODEL

Works that represent the current state of the art in NER generally start by representing words with pretrained word embeddings and embeddings that rely on surface form characters [3, 5]. These architectures feed the word representations to a bidirectional long short-term memory layer (Bi-LSTM, see Section 3.2.2) to produce fixed length vector representations of the context at every position. These representations are then used to disambiguate between the possible entities by decoding on trellis provided by a conditional random field (CRF) model.

In Chapter 3, we introduce such a model that has achieved state-of-the-art performance for several morphologically rich languages (MRLs). We have shown that using embeddings based on characters or linguistic properties of the word such as morphological features improves the performance compared to using only pretrained word embeddings. Even though this is a better approach for MRLs, the model in Chapter 3 require an external morphological disambiguator for every language of interest, a requirement which can be hard or even impossible for some languages to satisfy. This is especially true for agglutinative languages where there can be many roots and morphological tag sequences for a single word. Although there is an effort to provide a tool for POS tagging and lemmatization for many languages in a single format [108], it has been shown that there is a better approach for morphological tagging in terms of performance which can utilize the information in the context of the target word [41].

In this chapter, we propose a NER model which can exploit the morphological information by itself alleviating the external morphological disambiguator requirement. This model jointly learns the NER and morphological disambiguation (MD) tasks. We design our model so that any language with a mechanism that can provide a number of candidate morphological analyses for a word can utilize our joint model. This is

easier compared to providing disambiguated morphological analyses because systems that disambiguate morphological analyses are harder to build. Furthermore, we do not require the labels of each task to be present in the same dataset. One can easily train the part of the model that is responsible for the MD task in another -preferably larger- dataset and start with the pretrained model. Our main contribution is to show that jointly disambiguating morphological tags and predicting the NER tags results in an equivalent level of performance compared to using externally provided morphological tags.

We give a survey of related work on the subject in Section 2.2 and 2.3. Our proposed joint models are explained in Section 4.1. In Section 4.2, we describe our dataset that is derived from the Turkish dataset used in Chapter 3. After running the experiments described in Section 4.3, we observe that jointly training our model for NER and MD results in an increase in the NER performance.

## 4.1. Models

This section examines NER models. For this purpose, two base models are introduced:

(i) a Bi-LSTM based sequence tagger that predicts the correct NER tags with a CRF (Section 4.1.1)

(ii) a Bi-LSTM tagger that exploits the context representation to select the correct morphological analysis at the given position (Section 4.1.2)

These models are examined by introducing variations. First variation is the same model in Chapter 3 without the morphological embeddings (Section 3.3). Second variation is same as the state-of-the-art model in Chapter 3, which we call EXT_M_FEAT. The MD model has a single variation which is based on [41]. The joint models are formed of these two basic models combined in various ways (Section 4.1.3). In Section 4.3, we test our hypothesis by training these models by enabling or disabling various

components and input features [10] .

### 4.1.1. NER Model

We formally define an input sentence as $X = (x_1, x_2, \ldots, x_n)$ where each $x_i$ is a vector of size $l$ and the corresponding NER labels as $y_{\text{NER}} = (y_{\text{NER},1}, y_{\text{NER},2}, \ldots, y_{\text{NER},n})$. $x_i$ are then fed to a Bi-LSTM which is composed of two LSTMs (Section 3.2.2) treating the input forwards and backwards. The output of this Bi-LSTM at position $i$, $h_i$, is a vector of size $2p$ where $p$ is the size of the LSTM cell. Further, we transform $h_i$ through a fully connected layer $\texttt{FC}_{last}$ with `tanh` activations at the output to combine the left and right contexts into a vector of size $p$. This is followed by another fully connected layer to obtain a vector $s_i$ of size $K$, where $K$ is the number of the NER tags.

We follow a conditional random field (CRF) based approach to model the dependencies between the consequent tokens [97]. To do this, we take the vector $s_i$ at each position $i$ as the score vector of the corresponding word and aim to minimize the following loss function $\texttt{loss}_{\text{NER}}(X, y_{\text{NER}})$ for a single sample sentence $X$:

$$\texttt{loss}_{\text{NER}}(X, y_{\text{NER}}) = -\sum_{i=0}^{n} A_{y_i, y_{i+1}} - \sum_{i=1}^{n} s_{i,y_i} + \texttt{log}Z(X),$$
$$Z(X) = \sum_{y' \in \mathbb{Y}} \texttt{exp}\left(\sum_{i=0}^{n} A_{y_i, y'_{i+1}} + \sum_{i=1}^{n} s_{i,y'_i}\right)$$

where $A_{i,j}$ represents the score of a transition from tag $i$ to $j$, $\mathbb{Y}$ is the set of all possible label sequences. Using this model, we decode the most probable tagging sequence $y^*_{\text{NER}}$ as $\text{argmax}_{\tilde{y}_{\text{NER}}} \texttt{loss}_{\text{NER}}(X, \tilde{y}_{\text{NER}})$. We call this basic model the NER model [3] (see Figure 4.3).

In the remaining part of the section, we explain the details of the word representations used in this chapter.

[10] The code to replicate the experiment environment and the actual source code is published at https://github.com/onurgu/joint-ner-and-md-tagger

4.1.1.1. Representing words. As the default setting, we obtain word and character based embeddings as described below and combine them by concatenation. For the first component, we allocate a word embedding vector of size $d_w$ for every word in our dataset. This can be loaded from a pretrained word embeddings database as is done frequently in the literature, but we chose to learn the word embeddings during training. The second component is generated from the surface forms. We feed the character sequence of the word into a separate Bi-LSTM as described at the beginning of this section. However, instead of using the outputs of LSTM cells at each position, we just take the last and the first cell outputs of the forward and backward LSTMs and concatenate them (Figure 4.1). The resulting representation is two times the length of one character embedding length, $2d_c$. This second component is in turn concatenated with the first component to obtain a word representation vector $x_i$ of size $d_w + 2d_c$.

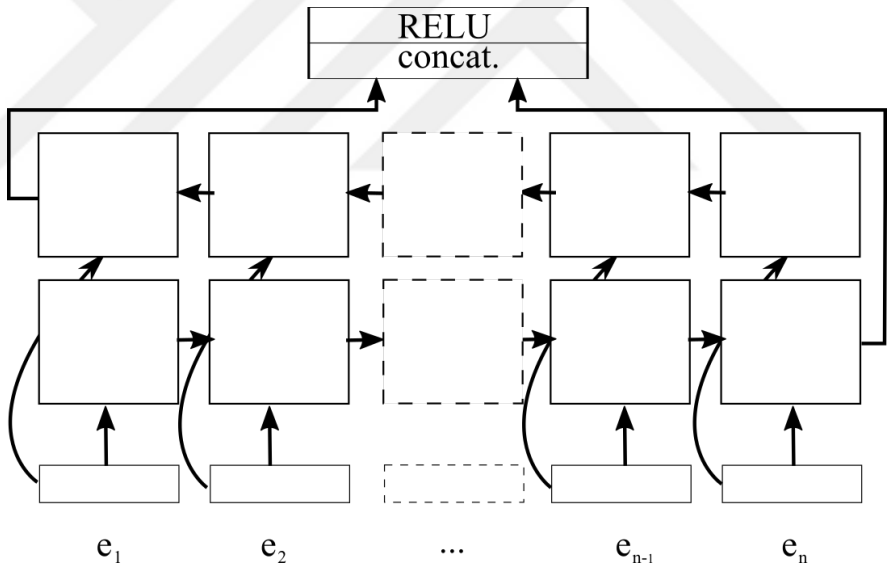



Figure 4.1. The Bi-LSTM model that generates word representations. Word or root surface forms and morphological tag sequences are treated using this architecture. The input sequence $(e_1, e_2, \cdots, e_{n-1}, e_n)$ can either be the character sequence of the whole word or the root of the word, or the tags in the morphological tag sequence. RELU unit is only active for roots and morphological tag sequences.

4.1.1.2. External morphological features. In order to compare our models with the proposed method in Chapter 3, we utilize the golden morphological analysis provided with the dataset in addition to the word and character based embeddings and call this model EXT_M_FEAT. The best approach in Chapter 3 treats the string form of

Analysis: `ev+Noun+A3pl+P3sg+Loc`

Transformation: `e v + N o u n + A 3 p l + P 3 s g + L o c`

Figure 4.2. Morphological analysis of Turkish word 'evlerinde' (in his/her houses) and its corresponding transformation.

a morphological analysis as a sequence of characters (Table 3.3), which is slightly different compared to the MD model in this chapter. EXT_M_FEAT model then applies the process depicted in Figure 4.1 to obtain morphological embeddings.

Using the sequence of characters of the morphological analysis instead of the sequence of morphemes might seem counterintuitive at the first glance. However, it is possible to give an explanation to show its benefits. For the sake of this explanation, the Turkish morphological analysis example and its corresponding transformation from Table 3.1 is presented in Figure 4.2. It has been argued that a benefit of treating morphological analyses as sequences of characters is that the information conveyed by the characters within the tags is shared among morphological embeddings. For example, in Turkish, the tags '`A3sg`' and '`A3pl`' indicate third person singular and third person plural where the leading two characters '`A3`' indicate third person agreement. This allows the model to represent the fragments of the tags which may improve the training performance. In this case, '`A3`' would represent the third person agreement independent of the singular or plural case. The resulting vector representation is thus of length $2d_m$ which is added to word and character based embeddings to obtain a word representation of $d_w + 2d_c + 2d_m$.

### 4.1.2. MD Model

In MD model, we are given a sentence $X$ in the same form as in the NER task, however we optimize the model to predict $y_{\text{MD}}$ where $y_{\text{MD},i}$ represents the correct morphological analysis out of the candidate morphological analyses for word $i$. As in the NER model, the MD model also employs a Bi-LSTM layer to obtain context representations when fed with the word representations $x_i$ (Figure 4.3). We define the candidate

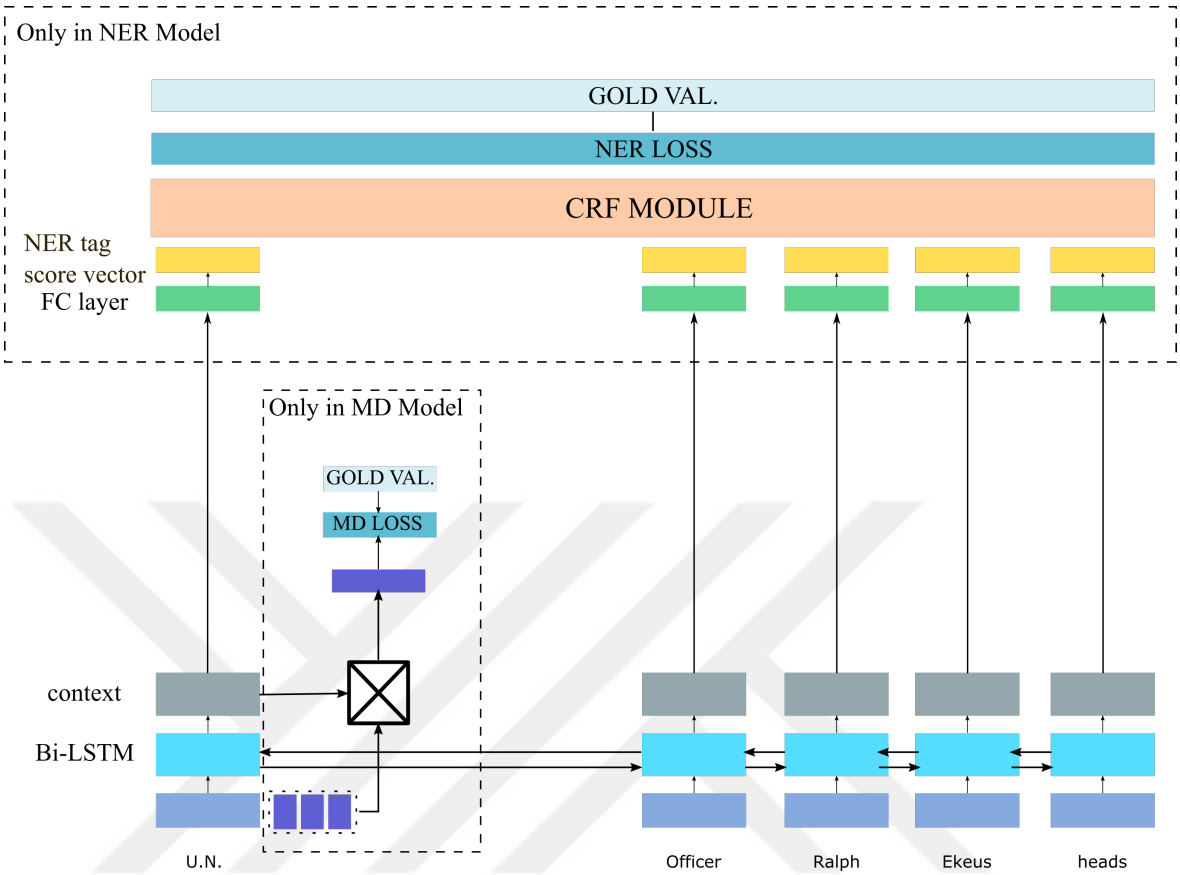



Figure 4.3. Our basic models: NER and MD. The portions of the model which are only active either for NER or MD models are indicated with dashed lines. The symbol ⊠ represents the selection of $\texttt{ma}_{ij^*}$.

morphological analyses for word $i$ as $\texttt{ma}_i = \{\texttt{ma}_{i,1}, \texttt{ma}_{i,2}, \cdots, \texttt{ma}_{i,j}, \cdots, \texttt{ma}_{i,K}\}$. To determine the correct morphological analysis, we examine each morphological analysis to extract the root surface form and the morpheme sequence and generate the representation $\texttt{ma}_{ij}$.

We design this approach to be generalizable to many morphological analysis output forms described in Section 2.2. We give an example from Turkish here: the unique analysis of the Turkish word "Moda'da" is "`Moda+Noun+Prop+A3sg+Pnon+Loc`" ('in Moda', which is a district in İstanbul) in which a common morpheme naming convention is used [7]. From this analysis, we determine the root as 'Moda' and the morpheme sequence as '`(Noun, Prop, A3sg, Pnon, Loc)`'. The root and the morpheme sequence are used to generate a representation as depicted in Figure 4.1. The RELU activation function [109] is also applied to the concatenation of the root and

morpheme sequence representations. We choose the resulting representations $r_{ij}$ and $\texttt{ms}_{ij}$ to be of two times the length of a morpheme embedding $d_m$. Furthermore, we add the root representation vector $r_{ij}$ and the morpheme sequence representation vector $\texttt{ms}_{ij}$ and apply hyperbolic tangent function (tanh), thus the morphological analysis representation $\texttt{ma}_{ij}$ is defined as follows $\texttt{tanh}(r_{ij} + \texttt{ms}_{ij})$.

We then select the morphological analysis $ma_{ij^*}$ by performing a dot product with the context vector $h_i$: $\texttt{ma}_{ij^*} = \text{argmax}_j\, h_i \cdot \texttt{ma}_{ij}$ when decoding. During training, the loss is calculated as

$$\texttt{loss}_{\text{MD}}(X, y_{\text{MD}}) = -\sum_{i=1}^{n} \texttt{log softmax}(\texttt{mscore}_i)$$

over a score vector $\texttt{mscore}_i$ such that $\texttt{mscore}_{ij} = \{h_i \cdot \texttt{ma}_{ij}\}$.

### 4.1.3. Joint model for NER and MD

We have experimented with three approaches for jointly learning NER and MD tasks. In this section, we explain the details of each approach.

4.1.3.1. Approach 1: Two losses. In this scheme, we employ a Bi-LSTM layer which is fed with word representations as in the basic models, NER and MD. We then use the same context $h_i$ to calculate the losses separately for NER and MD as explained in Sections 4.1.1 and 4.1.2. We call this joint model JOINT1 and show in Figure 4.4a. We then learn the model parameters to optimize $\texttt{loss}_{\text{JOINT1}}$

$$\texttt{loss}_{\text{NER}}(X, y_{\text{NER}}) + \texttt{loss}_{\text{MD}}(X, y_{\text{MD}}).$$

4.1.3.2. Approach 2: Enriching the context vector. As in the JOINT1 model, this model also calculates separate losses for each task and sums them to obtain a single loss to optimize. However, we additionally concatenate the selected morphological analysis

representation $\texttt{ma}_{ij^*}$ to $h_i$ before feeding it into the fully connected network with `tanh` outputs as described in Section 4.1.1. The model is shown in Figure 4.4a. The rationale of this concatenation is to facilitate information flow from the disambiguated morphological analysis. We call this model JOINT2. The loss function $\texttt{loss}_{\text{JOINT2}}$ of this model is then calculated similar to $\texttt{loss}_{\text{JOINT1}}$.

4.1.3.3. Approach 3: Multilayer and Shortcut Connections. Our most complicated model employs three Bi-LSTM layers instead of only one. We basically feed the output of the first layer $h_i^1$ to layer 2, the output of the second layer $h_i^2$ to layer 3. In addition to this, we transfer the word representation $x_i$ to all layer inputs and concatenate with $h_i^{level}$ to obtain $\overline{h}_i^{level}$. When processing to obtain the third layer's output $\overline{h}_i^3$, we also concatenate the selected morphological analysis representation $\texttt{ma}_{ij^*}$ to $h_i^3$ in addition to $x_i$. This is done to propagate the information gained from the disambiguated morphological analysis to the last layer of the network. We use the first layer's output $h_i^1$ when calculating $\texttt{mscore}_i$ as shown to be better for lower level NLP tasks such as POS tagging or chunking [51]. We call this model J_MULTI and depict in Figure 4.4b.

## 4.2. Data

To test our proposed model, we derived a new dataset based on the dataset that was employed in Chapter 3. This dataset is commonly used in the literature for the Turkish NER task [9]. The dataset contains sentences from the online edition of a Turkish national newspaper with NER labels. The creators of the dataset also provide a golden morphological analysis along with each word. However, golden morphological analyses in this dataset are sometimes erroneous. For example, words which are inflections of foreign words are usually problematic. An example is "Hillary'nin" which is the genitive case for the word "Hillary", i.e. 'Hillary's'. It is incorrectly labeled as if it is in nominal case. Also, when the surface form is a number in any noun case, like "98'e" ('towards 98') which is the dative case for the number ninety eight, the morphological analysis is almost always nominal. We believe that the reason for this is the incorrect handling of the quote character when preparing the original version.

In our study, we have first divided the training portion of the original dataset into training and development sets. We then augmented every word with a list of its all possible morphological analyses using a commonly used morphological analyzer [7]. These lists are used in morphological disambiguation component of our models. Unfortunately, the golden morphological analyses in about 5% of the word tokens were not found in these candidate analyses. This is probably due to the issues explained in the previous paragraph and to the changes in the morphological analyzer implementation. We built a tool to mitigate this issue semi-automatically.

The tool lists the most frequent contexts where a specific mismatch happens. The user selects the most suitable morphological analysis out of the candidates for each context. This is recorded as a solution to the mismatch. Using these solutions, we automatically corrected all contexts with a mismatch which has a solution in our solution database. Although we tried to give the utmost attention to selecting the best solution, some of our solutions might be problematic. Thus, we share the data, the scripts and the tool for academic use and examination[11] . Figure 4.5 shows the user interface of the tool that has been developed to execute this selection and correction process.

Unfortunately, there were still left a few hundred mismatches. As providing a solution for them required a lot of manual work and would only save 1-2 sentences for each, we just removed any sentence that contains any of these mismatches. This way, we have retained 25,511 out of 28,835 sentences in the original dataset for training, 2,953 of 3,336 for development and 2,913 of 3,328 for test. By this process, despite losing some of the sentences, we have built a new dataset with both the NER labels and the candidate morphological analyses which have correct golden labels.

### 4.2.1. Training

We implemented the model using the DyNet Neural Network Toolkit [110] in Python. The model parameters are basically the word embeddings, the parameters

[11]The data can be found at `https://github.com/onurgu/joint-ner-and-md-tagger`

of Bi-LSTMs, the weights of the fully connected layer $\texttt{FC}_{last}$, and the CRF transition matrix $A$. We trained by calculating the gradients of the loss for a batch of five sentences consisting of surface forms and its associated NER and/or MD labels and updated the parameters with Adam [111] for 50 epochs and reported the performance on test set of the model with the highest development set performance. We applied dropout [112] with probability 0.5 to the word representations $x_i$. To facilitate the reproducibility of our work, we also provide our system as a virtual environment[12] that provides the same environment on which we evaluated our system in an open manner.

### 4.3. Experiments and results

To test our approach, we train and evaluate every model for 10 times and report the mean F1-measure value for named entity recognition and accuracy for morphological disambiguation. This decreases the potential negative effects of random initialization of model parameters as shown in the literature [113]. However, to accomplish this, we set the parameter dimension sizes to 10 and do not employ pretrained word embeddings which is a necessary compromise given our limited computing resources.

The results are shown in Table 4.1. We see that the mean NER performance increases in joint models. Moreover, the JOINT2 model is performing better than just calculating two losses at the last layer as we did in the JOINT1 model. However, applying the Welch's t-test between the JOINT1 and JOINT2 runs does not strongly imply this difference ($p = .24$). Adding multiple Bi-LSTM layers to JOINT2 and obtaining J_MULTI also helped and achieved the best score among our joint models[13] . Employing Welch's t-test confirms the significance of this difference with other joint models, $p < .05$ for each pair.

To make a comparison with the method introduced in Chapter 3, we evaluated a

[12]You can obtain our implementation and find more information about how to use our virtual environment at `https://github.com/onurgu/joint-ner-and-md-tagger`.

[13]One can wonder whether this performance improvement could be due to an increase in the total number of parameters of the model. We saw that the increase is negligible as it only accounted for a 2% increase.

Table 4.1. Evaluation of our models for NER performance with our dataset. We report F1-measure results over the test portion of our dataset averaged over 10 replications of the training with the same hyper parameters.

| This work | |
|---|---|
| Model | Mean F1-measure |
| NER | 81.07 |
| JOINT1 | 81.28 |
| JOINT2 | 81.84 |
| J_MULTI | **83.21** |
| Previous work | |
| EXT_M_FEAT | **83.47** |

model where the golden morphological analysis in the corpus is represented as a vector and concatenated to the word representation $x_i$, namely EXT_M_FEAT (see Section 4.1.1.2). As one can see from Table 4.1, it achieved the best results compared to our joint models. However, we cannot confirm the difference between EXT_M_FEAT and J_MULTI models as the calculated $p$ is well above .05. Thus our best performing model J_MULTI is performing at a competitive level with an additional advantage of disambiguating the morphological tags while predicting the NER tags. This also serves as another confirmation to the hypothesis that employing linguistic information such as morphological features leads to an increase in the NER performance.

To evaluate the performance of morphological disambiguation, we have tested the MD performance of our models, which are trained with the training portion of our dataset, on the test portion of a frequently used dataset [114]. As can be seen from Table 4.2, we are very close to the state of the art MD performance even if we only trained with a low number of parameters as stated in the beginning of this section. We have to also note that in contrast with the NER task, the MD task did not enjoy a performance increase from joint learning. This might be due to the fact that NER can utilize the disambiguated morphological analysis of a word to predict the correct label, however a correctly predicted NER label does not contribute to the disambiguation of

the word's morphology.

Table 4.2. Evaluation of our models for MD performance. As in the NER evaluation, we report accuracies over the test dataset averaged over 10 replications of the training.

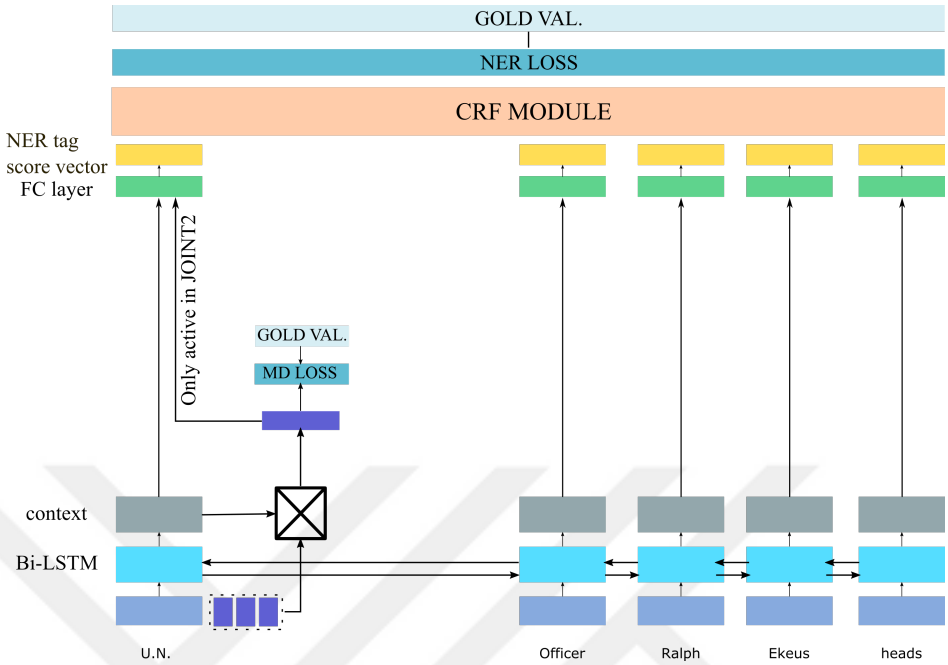
| This work | |
|---|---|
| Model | Mean Accuracy |
| MD | 88.61 |
| JOINT1 | 88.17 |
| JOINT2 | 86.86 |
| J_MULTI | 88.05 |
| Previous work | |
| [114] | 89.55 |
| [41] | **91.03** |

## 4.4. Conclusions

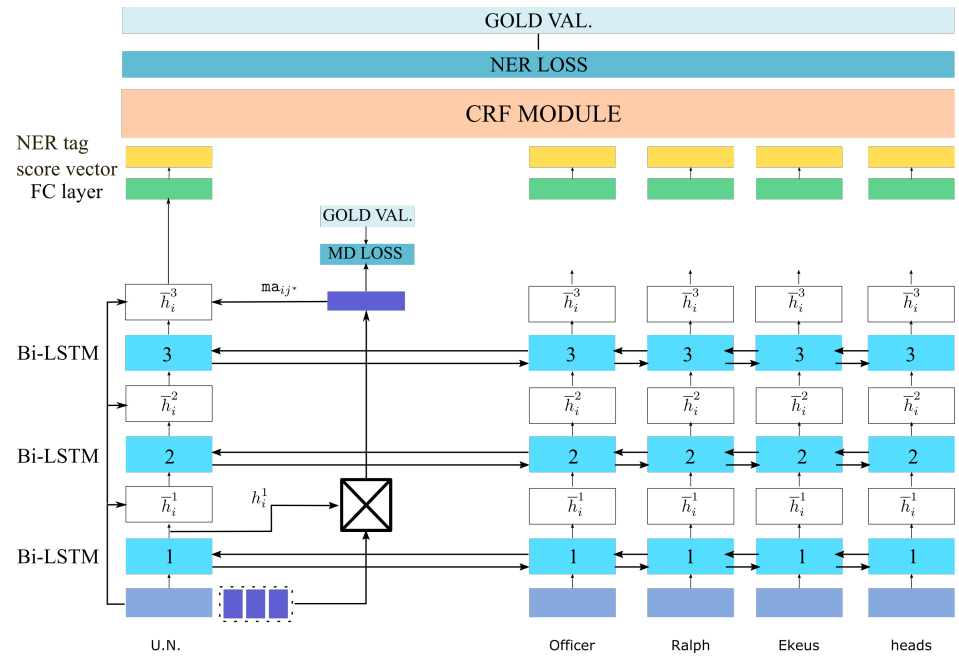
In this chapter, we introduced a joint model of NER and MD tasks that removes the need for external morphological disambiguators. Additionally, it is shown that the performance of this joint model can match the state-of-the-art NER model in Chapter 3. Achieving the same performance level without relying on an external morphological disambiguator is an improvement.

Moreover, the method is applicable to every language given that one can provide the candidate morphological analyses for a word, making this approach portable to many languages. We have also shown that joint learning itself leads to an increase in the NER tagging performance.

However, there is more work to do as we are still bound to language specific tools in obtaining the list of candidate morphological analyses. Generating the list of candidate analyses within the model, testing our hypothesis on other morphologically rich languages, and testing with models which have higher number of parameters are not addressed in this thesis and left for future work.




(a) (i) Model JOINT1: two losses for two tasks sharing a Bi-LSTM. (ii) Model JOINT2: We concatenate the selected morphological analysis' vector representation to the last layer's context vector.




(b) Model J_MULTI: We employ shortcut connections and two more Bi-LSTM layers.

Figure 4.4. Our joint models: (a) JOINT1 and JOINT2 models (b) J_MULTI model. The symbol ⊠ represents the selection of $\mathtt{ma}_{ij^*}$.

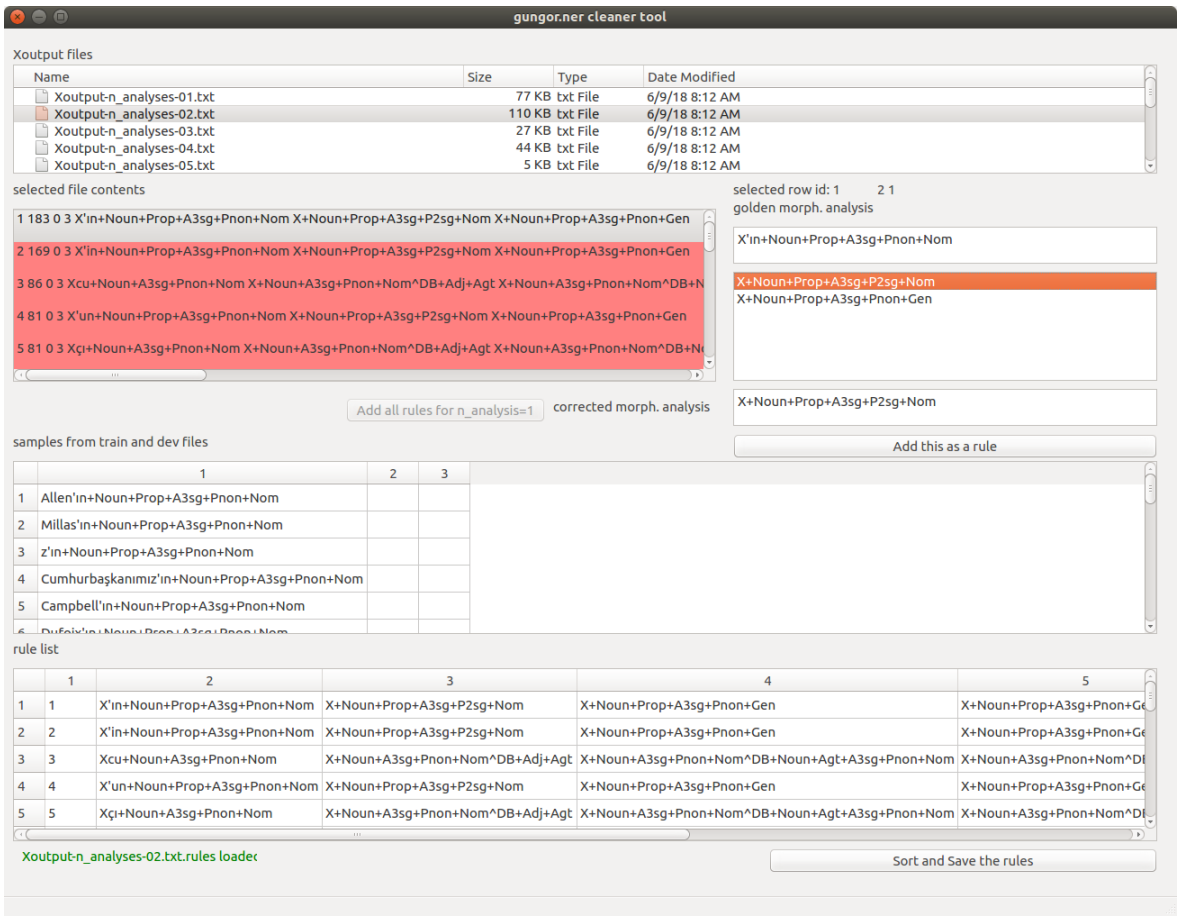



Figure 4.5. Graphical user interface of the tool that was developed to comb through the dataset to identify most frequent erroneous analyses and replace them with sensible ones.

# 5. EXPLAINING THE PREDICTIONS OF SEQUENTIAL TAGGERS

## 5.1. Introduction

In Chapter 3, we introduced a neural network based NER tagger which achieves state-of-the-art results for four morphologically rich languages. We show that the performance of a model that utilizes disambiguated morphological tags is superior to one that does not. We attribute the improvement in the performance to this modification. This explanation is satisfying only up to a degree. However, a more detailed explanation would be more convincing. For example, we would like to know which morphological features are most significant for determining that an entity is of type 'Location' in a single sentence. More specifically, we want to know which morphological features are relevant in determining the type of an entity in a given sentence as well as those that are generally significant in predicting a specific entity type.

Although explanations of this nature have not yet been proposed for NLP tasks, various approaches to provide explanations for machine learning predictions have been proposed [59,60,62,115,116]. One of the promising approaches to explain the outcome of a machine learning model is called Local Interpretable Model-Agnostic Explanations (LIME) [55], which proposes to explain a model's prediction based on the model's features. The given input sample is perturbed by randomly removing some features. The model's prediction function is employed to obtain probabilities corresponding to the perturbed versions of the input sample. LIME is based on the idea that the prediction probabilities of perturbed samples can be modeled by a linear model of features. The solution of the linear model is a vector of real values corresponding to the importance of each feature. Such vectors are considered to be valuable in assessing the quality of a model since they render the insignificant features evident, which are often the culprits in biased decisions.

In this chapter, we propose an extended version of LIME to handle any sequence-based NLP task in which a procedure for transforming the task into a multi-class classification problem can be constructed. This method utilizes *region*s which refer to the segments of the inputs that are directly related to the predictions, e.g. the tokens that cover a named entity in named entity recognition. The transformation procedure requires the probability of each prediction associated with a given region. Some models yield these probabilities as a part of their output. In other cases, access to the internals of the model are required to compute these probabilities. For example, for the sentiment classification task, typically a vector of class potentials is used to predict the sentiment of a sentence. This vector is used to calculate the probability of each sentiment type for the given sentence. For tasks with more complex labels, further computation may be required to calculate the probability of each prediction. For example, the probability of the entity tag for named entity recognition task is computed using the probabilities of the token-level tags. The prediction probabilities of each label in these perturbed samples are calculated using the transformation procedure specific to the task as detailed in Section 5.3.3 and 5.3.4.

The main aim of the proposed method is to provide a vector which indicates the strength and the direction of the impact of each feature. The first step is to observe the probability differences caused by the removal of each feature due to the perturbations. A linear regression model is used to relate these differences with the removed features. The solution of this linear model produces a list of weights corresponding to each feature. This list indicates the significance of each feature for a given prediction, which we consider to be an *explanation*.

To demonstrate the explanation method, we focus on the NER task using a NER tagger for the morphologically rich languages Turkish and Finnish [13]. The tagger requires all the morphological analyses of each token in the sentence to be provided to the model. Figure 5.1 shows a Turkish sentence and the potential morphological analyses for the named entity "Ali Sami Yen Stadyumu'nda" (meaning 'at the Ali Sami Yen Stadium' in English) that spans the tokens from the second to the fifth position. The model first predicts the correct morphological analyses, then uses to recognize the

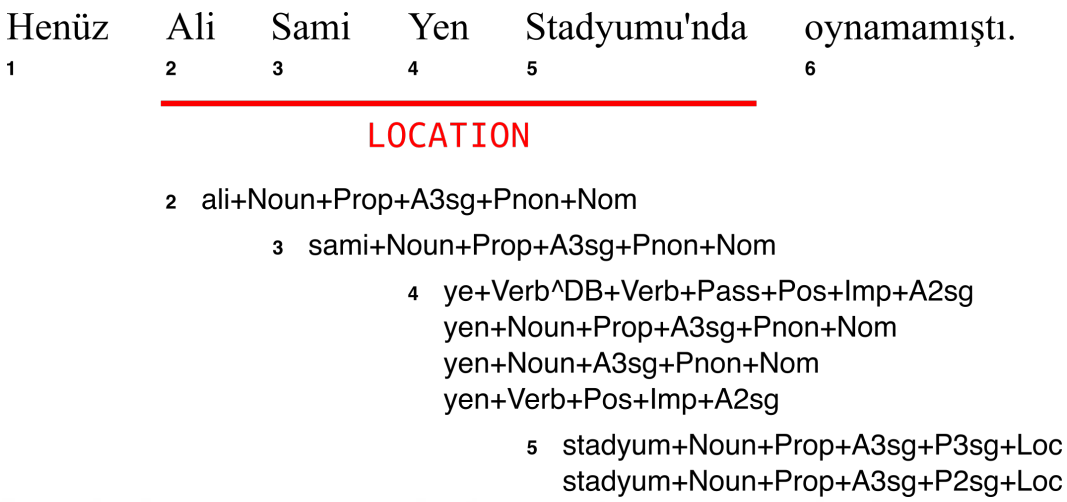



Figure 5.1. A Turkish sentence (translated as "She/he had not played at the Ali Sami Yen Stadium yet.") with one named entity tag and the possible morphological analyses of the tokens in the named entity.

named entities.

In Section 5.4, we provide quantitative and qualitative evaluations of the results of the explanation method. The quantitative evaluation compares the most influential morphological features in the predictions with those whose mutual information scores are the highest with respect to entity tags. The qualitative analysis is performed for both Turkish and Finnish by examining the morphological tags which are significant to several named entity tags.

The main contributions of the explanation related work are:

(i) a general method to explain predictions of any sequence-based NLP task by means of transforming them into multi-class classification problems,

(ii) a method to assess the impact of perturbations of input samples that relies on probability differences instead of the typical use of exact probabilities,

(iii) an encoding that distinguishes whether a feature absent in the perturbed sample was present in the original input, thereby capturing the knowledge of a removal operation,

(iv) a qualitative and quantitative evaluation of the proposed method for the NER task for the morphologically rich languages Turkish and Finnish, and

(v) an open source software resource to replicate all results reported in this chapter [117].

The work presented in this chapter is related to the current literature presented in Section 2.4. The remainder of this chapter is organized as follows: Section 5.2 explains the method that is the basis of our proposed explanation method which we introduce in Section 5.3. Section 5.4 presents the results of applying our method to Turkish and Finnish NER taggers. Finally, Section 5.5 summarizes the main takeaways and future directions.

## 5.2. The base explanation method: LIME

The Local Interpretable Model-Agnostic Explanations (LIME) [55] is a model-agnostic method for explaining the predictions of any machine learning model. It treats the model as a blackbox that produces a prediction along with an estimated probability. LIME belongs to the class of methods called additive feature attribution methods [57]. These methods yield a list of pairs composed of a feature and its impact on the prediction. This list is regarded as an explanation of the prediction based on the magnitude and the direction of the impact of each feature. Typically, these methods learn a linear model of the features to predict the expected probability of the prediction. The data samples required to train the linear model are obtained by perturbing the original input sample by removing a randomly selected feature.

In order to represent the features that are removed or retained during the perturbation, a binary vector $z$ that is mapped to the original input $x$ with a function $h$ is used. The mapping depends on the model to be explained. For example, if a model expects the input sentence $x$ to be in the bag-of-words form, $x$ consists of word and frequency pairs. In this case, the binary vector $z$ is composed of $z_i$s each of which indicate whether or not the $i$th word is retained. In other words, if $z_i$ is 1, word $i$'s bag of words frequency value remains the same as the frequency in the input sentence,

otherwise it is set to zero.

Additive feature attribution methods are generally defined as

$$g(z) = \phi_0 + \sum_{i=1} \phi_i z_i \tag{5.1}$$

where $z_i$ is the binary value that indicates whether feature $i$ is retained or not, $\phi_i$ is a value that indicates the importance of feature $i$, and $\phi_0$ is the bias. The function $g(z)$ is the outcome of the linear model that estimates the probability $f(x)$ which is obtained from the machine learning model. The following function is minimized to obtain the importance values $\phi_i$:

$$\underset{g}{\operatorname{argmin}}\ L(f, g, \Pi(x, z)) + \Omega(g) \tag{5.2}$$

where $f$ is the probability function of the model, $\Pi(x, z)$ is the local weighting function and $\Omega(g)$ constrains the complexity of $g$. For example, to explain a text classification model, one might set $\Pi(x, z)$ to an exponential kernel with cosine distance between $x$ and $z$. Any function that satisfies the distance constraints, namely the non-negativity, zero distance if $x = z$, symmetry, and the triangle inequality ($d(x, z) \leq d(x, y)+d(y, z)$), may be used for $\Pi(x, z)$. A reasonable choice for $\Omega(g)$ is a function that returns the number of words in the vocabulary. Accordingly, the loss function $L$ is defined as the sum of the squared errors weighted by $\Pi(x, z)$:

$$L(f, g, \Pi(x, z)) = \sum_{x, z=h(x)} \Pi(x, z)(f(x) - g(z))^2. \tag{5.3}$$

### 5.3. Explaining sequence-based NLP tasks

This section introduces a method for explaining specific predictions of models trained for sequence-based NLP tasks. Essentially, the method provides explanations about which part of the input impacts the prediction of a given neural model. The

method produces an explanation vector of scores that indicate the impact of the features used by the model. This vector can be utilized in offering an explanation to the user of the model's prediction. For example, a model trained for classifying the sentiment of a sentence may rely on features such as the specific words that occur in the sentence, the position and the number of punctuation marks in the sentence, or the content of the fixed-size vector representations pretrained for each word. In this sentiment classification task, the user should be suspicious of a model if words that have clear negative sentiment are effective in a positive sentiment prediction.

### 5.3.1. Defining NLP tasks

We define NLP tasks as *processes* that transform input consisting of a sequence of tokens along with a set of features into a sequence of labeled tokens. Figure 5.2 provides an overview of NLP processes. In this chapter, we denote the input with $X$, the tokens for input with $T$ and for output with $T'$, the number of tokens for input with $N_t$ and for output with $N_o$, the output labels with $Y$, and the number of output labels with $N_y$. These *processes* are implemented by *models*. Each *model* has a *prediction function* that maps $X$ to $T'$ and $Y$. The model architecture determines the size and the contents of the feature sets $F$. An example for a type of feature could be the word embedding that corresponds to a token in $T$. This is a flexible definition that applies to nearly all NLP tasks.

Some NLP tasks and their corresponding parameters are shown in Table 5.1. The sentiment classification task can be associated with the question: "Is the sentiment of the sentence X positive?". The expected output is simply "Yes" or "No". In this case, there are no token outputs, thus $N_o = 0$ and the cardinality of the label space is two (or $|Y| = 2$). The word sense disambiguation task can be expressed with the question "What is the sense of the word $X$?" whose answer is one of the expected word senses. In this case, again, $N_o = 0$ while $N_y = 1$, but the cardinality of the output label space is the number of possible senses. The NER task can be expressed as a mapping from each input token to a named entity tag. As such, $N_o = 0$, $N_y = N_t$ and the cardinality of the output label space is the number of token tag sequences of length $N_y$. Machine

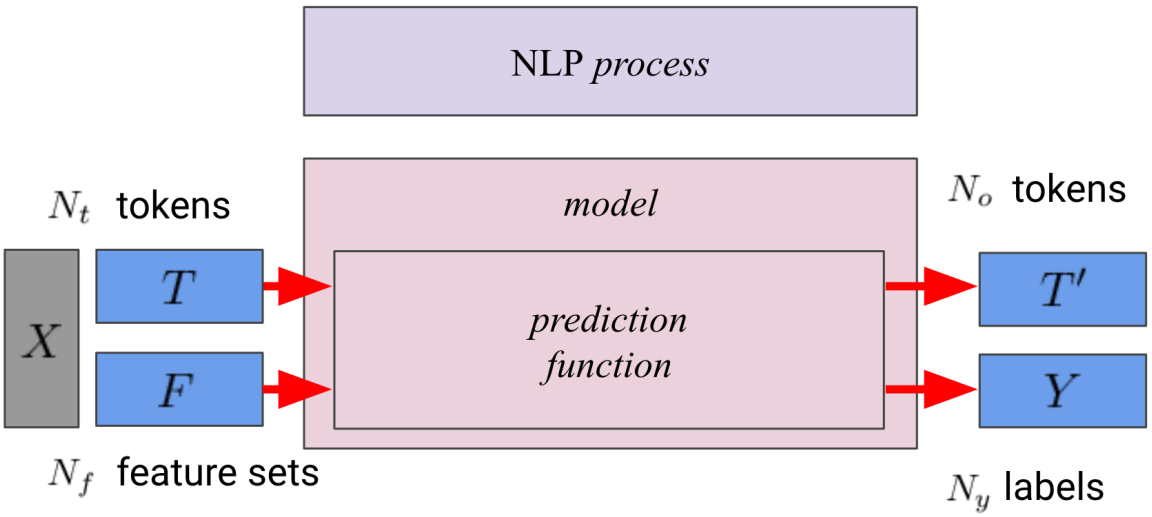



Figure 5.2. Relations between processes, models, input, and output in the scope of our explanation method.

translation can also be defined by this scheme by setting $N_t > 0$, $N_o > 0$, $N_y = 0$, and the cardinality of the output token space to the number of token sequences of length $N_o$. In another case, the task may require both output tokens and output labels, like in morphological disambiguation which we give an example in Table 5.1. This scheme is flexible enough to express models that employ features concerning the whole sentence. For example, an alternative version of the example for sentiment classification could have a single feature set for the whole sentence (e.g. sentence embedding). The parameter configuration for this case would be $N_t > 0, N_f = 1, N_o = 0$, and $N_y = 1$.

### 5.3.2. EXSEQREG: Explaining sequence-based NLP tasks with regions

This section describes the proposed framework for explaining neural NLP models. For illustration purposes, we use the NER tagger we introduced in Chapter 4 as a use case. We describe our method using a set of variables along with the indices $i$, $t$, $j$, and $k$ (see Table 5.2). These indices are used to refer to input sentence $X_i$ as the $i$th sentence in the dataset, feature set $F_{it}$ corresponding to $t$th token in sentence $i$, and label $Y_{it}$ corresponding to $t$th token in sentence $i$. Figure 5.3 depicts an example that utilizes these variables. For the NER tagger, $Y_{it}$ is the token-level named entity tag, e.g. 'B-PER', 'I-PER', 'I-LOC', and similar. The input to the NER tagger consists of

Table 5.1. Selected examples of NLP tasks that can be covered by the method.

| **Task** | $T$ | $F$ | $T'$ | $Y$ |
|---|---|---|---|---|
| Sentiment classification (model 1) | "Great music!" $N_t = 2$ | A feature set for each token, $N_f = 2$ | $N_o = 0$ | 'YES' $N_y = 1$ |
| Sentiment classification (model 2) | "Great music!" $N_t = 2$ | A feature set for the whole sentence, $N_f = 1$ | $N_o = 0$ | 'YES' $N_y = 1$ |
| Word sense disambiguation | "We bought gas for the **car**." $N_t = 6$ | A feature set for each token, $N_f = 6$ | $N_o = 0$ | *Sense* = automobile $N_y = 1$ |
| Named Entity Recognition (in Turkish) | "Henüz **Ali Sami Yen Stadyumu** taşınmamıştı.", $N_t = 6$ | A feature set for each token, $N_f = 6$ | $N_o = 0$ | "O B-LOC I-LOC I-LOC I-LOC O" $N_y = N_t = 6$ |
| Machine translation (from Turkish to English) | "Henüz Ali Sami Yen Stadyumu taşınmamıştı.", $N_t = 6$ | A feature set for each token, $N_f = 6$ | "Ali Sami Yen Stadium was not relocated yet." $N_o = 8$ | $N_y = 0$ |
| Morphological disambiguation (in Turkish) | "Henüz Ali Sami Yen Stadyumu taşınmamıştı.", $N_t = 6$ | A feature set for each token, $N_f = 6$ | "Henüz Ali Sami Yen Stadyum+u taşın+ma+mış +tı" $N_o = 6$ | "Henüz Ali Sami Yen Stadyum+Acc taşın+Neg +Past-Part+Past", $N_y = 6$ |

Table 5.2. Summary of variables used by the explanation method.

| **Variable** | **Description** |
|---|---|
| $i, X$ | input sequence index and input sequence |
| $t, T$ | token index and token sequence |
| $f, F$ | feature and set of features |
| $Y$ | label sequence |
| $j, R$ | region index in sentence and set of regions |
| $\pi, \Pi$ | perturbed sentence and set of perturbed sentences |
| $k, K$ | label $k$ and number of labels |
| $e_{ij}^{k\bigstar}$ | explanation vector for region $j$ in sentence $i$ for label $k$ |

the morphological analyses, the word embeddings and the surface forms of the tokens. The NER tagger exploits the information conveyed by the morphological tags within the analyses.

The method proposes the concept of region, which is used to refer to a specific part of the input sentence. For example, for the NER task, regions refer to named entities which may span several consecutive tokens. Regions are used to associate features and predictions related to a segment of the input. Figure 5.3 shows a Finnish sentence with two regions marking named entities, one of type 'PRODUCT' and the other of type 'PERSON'.

We define an explanation vector $\mathbf{e}_{ij}^{k\bigstar}$ to explain the prediction of label $k$ for the $j$th region of sentence $i$. This vector's length equals the number of unique features in the model. The regions are denoted by a sequence of integers that give the positions of the tokens belonging to the region. For the NER problem, $k$ is the named entity tag and the number of unique features is equal to number of unique morphological tags in the model. The values of the dimensions of $\mathbf{e}_{ij}^{k\bigstar}$ represent the impact of their

corresponding features.

A full example of the NER task is depicted in Figure 5.3 where $T_i$ is the sequence of words in sentence $X_i$ and $N_t$ is the number of words. There are no output tokens for this task, thus $N_o = 0$. There is a label and a list of morphological analyses corresponding to each input token, making $N_y = N_t = N_f$. The regions which contain each named entity are denoted as $r_{ij}$. As shown in the figure, $r_{i1}$ spans the first three tokens, 'Amazon Web Servicesin', and is labeled with named entity tag 'PRO' which signifies a product name. This can be seen by observing the $Y_{it}$ values which are the token-level named entity tags. The lower part of the figure lists all possible morphological analyses for each token $t$ and its feature sets $F_{it}$ originating from these lists. The union of every $F_{it}$ in the region $r_{i1}$ is denoted as $\mathbb{F}_{i1}$. The explanation vector $\mathbf{e}_{i1}^{\texttt{PRO}\bigstar}$ in the lower right part of the figure contains a real value for each feature in $\mathbb{F}_{i1}$. The diagram indicates that 'Case=Nom' contributes positively to the 'PRO' prediction. On the other hand, the presence of the 'Number=Sing' and 'Case=Gen' morphological tags is expected to decrease the probability of the 'PRO' named entity tag. This is a simple example where the explanations can be viewed as the degree to which a morphological tag is responsible for identifying a specific named entity tag.

The remainder of this section describes the main steps required to calculate $\mathbf{e}_{ij}^{k\bigstar}$:

(i) Perturbing $X_i$ to obtain a set of sentences $\Pi_i$ (Section 5.3.3).

(ii) Calculating probability changes corresponding to all regions $r_{ij}$ of $X_i$ using sentences in $\Pi_i$ (Section 5.3.4).

(iii) Defining and solving a special regression problem corresponding to every region $r_{ij}$ in every perturbed sentence in $\Pi_i$ (Section 5.3.5).

### 5.3.3. Perturbation

NLP tasks are divided into several classes according to their region types. The widest regions span entire sentences, such as in the case of sentiment classification. The regions within sentences may be contiguous or not. For example, the NER task is

almost always concerned with contiguous regions but the co-reference resolution task or the multi-word expression detection task is usually characterized by noncontiguous regions.

In all of these cases, the $j$th region in sentence $i$ is denoted as $r_{ij}$ and represented by a sequence of integers that correspond to the positions of the tokens in the region. We define $R_i$ as the set of all regions $r_{ij}$ in $X_i$. For every $r_{ij}$ in $R_i$, we perturb $X_i$ by only modifying the features that are found in that region. A region $r_{ij}$ in the NER task is represented with a sequence of integers, i.e. $(start, \ldots, end)$ where *start* and *end* are the first and last position indices of the words in the region. For example, in the Turkish sentence "Henüz Ali Sami Yen Stadyumu'nda oynamamıştı", there exists a single region which spans the words through the second to the fifth, i.e. $(2, 3, 4, 5)$.

The set of features subject to perturbation in region $r_{ij}$ is defined as $\mathbb{F}_{ij} = \bigcup_{t \in r_{ij}} F_{it}$. We perturb $X_i$ by independently removing each feature $f \in \mathbb{F}_{ij}$ from $X_i$ to obtain $\pi_{ij} = \{\texttt{remove}(X_i, j, f) : f \in \mathbb{F}_{ij}\}$. The expression $\texttt{remove}(X_i, j, f)$ denotes a sentence originating from sentence $X_i$ where all instances of feature $f$ are removed from all $F_{it}$ in region $r_{ij}$. The unperturbed version of $X_i$ is denoted as $\pi^{\emptyset}_{ij}$. Figure 5.4 shows the change in $\mathbb{F}_{ij}$ corresponding to each perturbed version $\pi^{r}_{ij}$. To form $\pi^{1}_{i1}$, the morphological tag 'Number=Sing' is removed from the morphological analyses of tokens 1, 2, and 3 (e.g. 'Amazo| ~NOUN~N~Case=Gen|Number=Sing' to yield 'Amazo| ~NOUN~N~Case=Gen'). The collection of $\pi_{ij}$s results in a set $\Pi_i$ consisting of at most $\sum_{j'=1}^{|R_i|} |\mathbb{F}_{ij'}|$ perturbed sentences, which is at most $|R_i| \times |M|$ where $M$ is the set of unique features in the model. For the case of the NER task $M$ is relatively low. For cases where $M$ is very high, the number of perturbations could be computationally overwhelming. For example, the number of features that are constructed combinatorially from input segments becomes very large as sentence lengths increase. In such cases, the feature to be removed could be selected in a uniformly random manner from $\mathbb{F}$. This is repeated for several times to form a set of perturbed samples with a feasible size.

Eventually, a set of perturbed samples $\pi_{ij}$ for each region $r_{ij}$ is obtained to be

used as input to the prediction function of the model.

### 5.3.4. Calculating probabilities

In this step, we seek to obtain a matrix of label prediction probabilities $P_{ij}$ where the $r$th row corresponds to the $r$th perturbed version of $X_i$ in $\pi_{ij}$ ($\pi^r_{ij}$). Each row of $P_{ij}$ is a vector $p^r_{ij}$ of length $K$ where each dimension corresponds to a label $k$ of the task at hand. Thus the size of $P_{ij}$ is $|\pi_{ij}| \times K$.

Depending on the task and the model, $p^r_{ij}$ might be computed by the model itself. For instance, a sentiment classification model might yield the probability of the positive label directly. On the other hand, it might be necessary to compute $p^r_{ij}$ using some output of the model. Some models include a component which indirectly corresponds to the prediction probability of each label $k$ in a region $r_{ij}$. For example, our NER tagger in Chapter 4 aims to find the contiguous sequence of IOBES tokens referring to named entities with the highest probability but it does not directly output the probability. However, special transformation mechanisms may be designed to obtain these probabilities. We give a detailed example in the following paragraphs.

For tasks that do not output the prediction probabilities, we need a mechanism for transforming them into multi-class classification problems to provide an explanation for the prediction in region $r_{ij}$. For the NER task, the IOBES tags in the region must be transformed to named entity tags. The transformation procedure selects paths satisfying the following regular expression "S-TAGTYPE | B-TAGTYPE,[I-TAGTYPE]*,E-TAGTYPE | O+". The resulting path list is filtered so that it only includes paths with a single entity. We omit paths that result in multiple entities or paths that are invalid (e.g. starting with a 'I-' prefix) in the region as the trained model consistently attaches very low probabilities to such cases. For other NLP tasks, one should start with enumerating all the possible prediction outcomes in a given region $r_{ij}$. If the number of total outcomes in a region is very high, it is advised to omit the outcomes which are expected to have very low probabilities. After this filtering, each remaining outcome is considered as a label.

Figure 5.5 shows the correct sequence of IOBES tags for a Turkish named entity tag 'LOC'. This is one of the $13^4$ possible sequences. The number of possible sequences is calculated by multiplying the number of possible token-level tags at each token position $t$. In this case, the total number of possible sequences is calculated as $(4 * K + 1)^N$ where $K$ is the number of entity tags and $N$ is the number of tokens.

After this transformation procedure, the NER task which is originally a sequence tagging problem is reduced to a classification problem with $K$ classes. The NER tagger uses score variables $s_{t,o}$ to predict IOBES tags ($o$) for each position $t$ (Figure 5.5). During normal operation, the NER tagger feeds these scores to a Conditional Random Field (CRF) layer. The CRF layer treats these scores as token-level log-likelihoods and uses the learned transition likelihoods to choose the most probable path (Section 4.1). For the purposes of the explanation method, we define the probability of the sequence corresponding to the entity tag $k$ in the named entity region $r_{ij}$ as

$$p^r_{ij}(k) = P(k|\pi^r_{ij}) = \frac{\exp(\texttt{score}(\pi^r_{ij}, k))}{Z^r_{ij}}$$

where $\texttt{score}(\pi^r_{ij}, k)$ is the total score of entity tag $k$ and $Z^r_{ij}$ is $\sum_{k'} \exp(\texttt{score}(\pi^r_{ij}, k'))$. We also define the same probability for region $r_{ij}$ in the unperturbed sentence $X_i$ and refer to it as $p^{\emptyset}_{ij}$.

### 5.3.5. Computing importance values

The previous steps of the method produce a $\pi_{ij}$ and $P_{ij}$ for each region $r_{ij}$. The final step aims to produce an explanation for every label $k$ for every region $r_{ij}$. An explanation $e^k_{ij}$ for label $k$ of region $j$ in sentence $X_i$ is a vector which contains one dimension for each feature in the model. Each element of $e^k_{ij}$ indicates the impact of a feature $m$ from $M$ for predicting label $k$, where $M$ is the set of features used in the model. Note that a region in a given sentence $X_i$ is not always related to all features (e.g. most of the morphological tags do not exist in all regions in the NER task), thus the number of features $|\mathbb{F}_{ij}|$ related to region $r_{ij}$ is usually smaller than $|M|$ and

$|M| - |\mathbb{F}_{ij}|$ dimensions are guaranteed to be zero.

We add the original sentence $X_i$ to $\pi_{ij}$ together with $|\mathbb{F}_{ij}|$ perturbed sentences to obtain a set of $|\mathbb{F}_{ij}| + 1$ sentences. We first form a matrix $\mathbb{C}_{ij}$ of size $(|\mathbb{F}_{ij}| + 1) \times |M|$ where the $r$th row corresponds to $\pi_{ij}^r$ if $r \leq |\mathbb{F}_{ij}|$. The last row of $\mathbb{C}_{ij}$ corresponds to the unperturbed version of $X_i$. Every row of $\mathbb{C}_{ij}$ is composed of ones, minus ones, and zeros signifying whether the feature that corresponds to the $m$th position was i) present and retained, ii) present and removed, and iii) was not present in input, respectively. We prefer this scheme to using a one or zero to indicate the presence or absence of a feature in the perturbed sentence, since we would like to penalize the features that were present and removed.

Secondly, we form a matrix $\Delta\mathbb{P}_{ij}$ of size $K \times (|\mathbb{F}_{ij}| + 1)$ where the $r$th column of the first $|\mathbb{F}_{ij}|$ columns is equal to $p_{ij}^r - p_{ij}^{\emptyset}$. The last column is set to $\vec{0}$ as there is no perturbation in the original sentence. In other words, the entry $(k, r)$ of $\Delta\mathbb{P}_{ij}$ contains the difference induced in the probability of predicting label $k$ after the perturbation described by the row $r$ of $\mathbb{C}_{ij}$.

The matrices $\mathbb{C}_{ij}$ and $\Delta\mathbb{P}_{ij}$ are then combined in the loss function of ridge regression which employs regularization on the explanation vector

$$||\Delta\mathbb{P}_{ij}(k,:) - \mathbb{C}_{ij}e_{ij}^k||_2^2 + ||e_{ij}^k||_2^2 \tag{5.4}$$

and minimized with respect to $e_{ij}^k$. We use notation $A(i,:)$ to refer to the $i$th row of matrix $A$. We call the corresponding solution as $\mathbf{e}_{ij}^{k\bigstar}$. We give the pseudo-code of the method in Algorithm 5.6. In contrast to Equation 5.3 in Section 5.2, we do not use a distance function in this formula because every perturbed sentence is assumed to be at the same distance from the original sentence.

For illustration purposes, consider a very simple task with two features (*Non-emotional* and *Emotional*) and three tags (*Positive*, *Negative*, and *Neutral*). Thus, $|M| = 2$ and $K = 3$. Let's further assume that the sentence $i$ consists of a single region

which spans the whole sentence and both features are present in this region. So, there is a single region $r_{i1}$ and $|\mathbb{F}_{i1}| = 2$. Thus $\Delta\mathbb{P}_{i1}$ is of size $3 \times (2+1)$ and $\mathbb{C}_{i1}$ is of size $(2+1) \times 2$. Let's choose the entries of these matrices so that we observe that the probability of predicting '*Positive*' label for the sentence increases when

- feature *Non-emotional* was present and removed, and
- *Emotional* was present but not removed.

This is represented in the first row of $\mathbb{C}_{i1}$ (i.e. $[-1, 1]$) and in $\Delta\mathbb{P}_{i1}(1,1)$ (i.e. 0.3) in the following equations. We chose the other values so that the probability of predicting '*Positive*' decreases (i.e. $-0.1$) when the '*Emotional*' feature is present and removed, and the '*Non-emotional*' feature is present but not removed.

$$\Delta\mathbb{P}_{i1}(1,:) = \begin{bmatrix} 0.3 & -0.1 & 0 \end{bmatrix}^{\mathrm{T}} \tag{5.5a}$$

$$\mathbb{C}_{i1} = \begin{bmatrix} -1 & 1 \\ 1 & -1 \\ 1 & 1 \end{bmatrix} \tag{5.5b}$$

According to the definitions above, we can define the explanation vector for label *Positive* as

$$\mathbf{e}_{i1}^{1\bigstar} = \underset{e_{i1}^{1}}{\operatorname{argmin}} ||\Delta\mathbb{P}_{i1}(1,:) - \mathbb{C}_{i1} e_{i1}^{1}||_2^2 + ||e_{i1}^{1}||_2^2. \tag{5.6}$$

When we solve it, we obtain

$$\mathbf{e}_{i1}^{1\bigstar} = \begin{bmatrix} -0.08 & 0.08 \end{bmatrix}^{T} \tag{5.7}$$

which can be interpreted as the '*Non-emotional*' feature has a negative impact on the prediction of label '*Positive*', whereas the '*Emotional*' feature has an opposite impact.

To apply this method to the NER task, the $\Delta\mathbb{P}$ and $\mathbb{C}$ matrices are set up in accordance with the parameters. For example, the Finnish NER dataset requires $K$ and $|M|$ to be set to 10 and 89, respectively. These values are 3 and 181, respectively, for the Turkish NER dataset. The method provides the explanation vectors $\mathbf{e}_{ij}^{k}$ for every entity tag $k$ for every region $r_{ij}$ for every sentence $X_i$. The values in the $m$th dimension of this vector indicate the magnitude and direction of the $m$th morphological tag's impact on the entity tag $k$ for region $r_{ij}$.

## 5.4. Analysis

To assess the proposed method, two NER taggers were trained for Turkish and Finnish with appropriate datasets as described in Section 4.2.1 of this thesis. We followed the same training regime except that we employed 100 dimensional embeddings instead of 10.

As the NER tagger jointly models the NER task and the morphological disambiguation (MD) task, two data sources are required for each language (Table 5.3). For Finnish, we used a NER dataset of 15,436 sentences for training NER related parts of the model [104, 118]. For MD related parts, we used 172,788 sentences from the Universal Dependencies dataset [119] using the modified version [120] of the UdPipe morphological tagger instead of the original version [121] because the NER tagger proceseses all possible morphological analyses instead of just the disambiguated morphological analysis.

For Turkish, we used our extended version of the most prevalent Turkish NER dataset which is explained in Section 4.2. The dataset was extended so that it includes all morphological analyses instead of just the disambiguated analysis.

### 5.4.1. Results

The evaluation of explanation methods remains an active research area. Several approaches have emerged ranging from manual assessments to qualitative and qualita-

Table 5.3. The hyperparameters and region-related statistics for Turkish and Finnish NER datasets.

| Language | $M$ | $K$ | # regions | $r$ per sentence | ORG | TIT | PER | TIM | LOC | DATE | PRO | MISC | EVENT | OUTSIDE |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Turkish | 181 | 3 | 22,364 | 1.85 | 6,268 | - | 9,318 | - | 6,778 | - | - | - | - | - |
| Finnish | 89 | 10 | 25,578 | 2.27 | 9,101 | 631 | 2,229 | 5,148 | 2,040 | 956 | 4,467 | 908 | 93 | 5 |

tive methods with no consensus as of yet [122]. To evaluate the explanation vectors, we utilize three metrics based on the mean of standardized importance values ($\hat{\mu}_k$), the distribution of standardized importance values across the corpus ($\hat{\mathbb{E}}^k(m)$), and the mutual information gain ($MI_{k,m}$), which are defined in the next section.

We evaluate the explanations as follows:

(i) As a form of qualitative evaluation, the average importance values for each morphological tag $m$ (denoted as $\hat{\mu}_k(m)$) are ranked in order of significance. This ranking is compared with the expected ranking based on our knowledge of the language features.

(ii) We visually inspect the importance values of the morphological tags using $\hat{\mathbb{E}}^k(m)$.

(iii) We determine the morphological tags that are important for all entity tags using $\hat{\mu}_k(m)$.

(iv) As a quantitative approach, we calculate the mutual information gain between each morphological tag ($m$) and entity tag ($k$) denoted as $MI_{k,m}$ and rank the morphological tags according to this metric to observe the number of matches with the results of the proposed method.

(v) Finally, we designed an experiment to observe whether removing the higher ranked morphological tags more significantly decreases the performance in com-

parison to the removal of lower ranked tags.

These approaches are used to evaluate the computed explanation vectors for Turkish and Finnish. The code for computing the metrics are shared with the research community on a public website [117].

### 5.4.2. Metrics

We process the Finnish and Turkish corpora so that $F_{it}$ is the union of all morphological tags in all possible morphological analyses of the $t$th word in the $i$th sentence. We then employ the explanation method given in Algorithm 5.6 using the corresponding NER tagger to obtain the explanation vectors $\mathbf{e}_{ij}^{k\bigstar}$ of size $|M|$ for every named entity region $r_{ij}$ in every sentence $X_i$ by solving Equation 5.4.

We then calculate

$$\mu_k = \frac{\sum_{i,j} \mathbf{e}_{ij}^{k\bigstar}}{N_k}, \tag{5.8a}$$

$$\sigma_k = \sqrt{\frac{\sum_{i,j} (\mathbf{e}_{ij}^{k\bigstar} - \mu_k)^2}{N_k}}, \tag{5.8b}$$

$$\hat{\mathbf{e}}_{ij}^{k\bigstar} = \frac{\mathbf{e}_{ij}^{k\bigstar} - \mu_k}{\sigma_k} \tag{5.8c}$$

$$\hat{\mu}_k = \frac{\sum_{i,j} \hat{\mathbf{e}}_{ij}^{k\bigstar}}{N_k}. \tag{5.8d}$$

where $N_k$ is the number of named entity regions labeled with the named entity $k$ in the corpus. The $\mu_k$ and $\sigma_k$ are vectors of size $|M|$ where each dimension is the mean and variance of the importance values of the corresponding morphological tag $m$. The $\hat{\mathbf{e}}_{ij}^{k\bigstar}$ is obtained by standardizing the values using the mean and variance of $\mathbf{e}_{ij}^{k\bigstar}$. The $\hat{\mu}_k$ is the standardized version of $\mu_k$.

Furthermore, we define

$$\hat{\mathbb{E}}^k(m) = \left[\hat{\mathbf{e}}_{ij}^{k\bigstar}(m) : \forall i, j\right] \tag{5.9}$$

as a vector containing all values in the $m$th dimension of all explanation vectors in all regions $r_{ij}$ with label $k$. This variable is useful in analyzing the distribution of standardized importance values across the corpus.

The metric $MI_{k,m}$ is defined to quantify the information given by a morphological tag $m$ to predict an entity tag $k$. To calculate this metric, a pair of vectors $(L_k, \Phi_{k,m})$ are defined. Each vector has N dimensions which correspond to the total number of regions in the corpus. Each dimension in $L_k$ is set to 1 if the region is labeled with entity tag $k$, or to 0 otherwise. Likewise, each dimension of $\Phi_{k,m}$ is set to 1 if the region contains morphological tag $m$, or to 0 otherwise. Using these vectors, the mutual information score is computed for each pair of $k$ and $m$:

$$MI_{k,m} = \sum_{j=0}^{1}\sum_{j'=0}^{1} \frac{|L_k(j) \cap \Phi_{k,m}(j')|}{N} log \frac{N|L_k(j) \cap \Phi_{k,m}(j')|}{|L_k(j)||\Phi_{k,m}(j')|} \tag{5.10}$$

where $L_k(j)$ and $\Phi_{k,m}(j)$ denote the set of indices that are set to $j$.

### 5.4.3. Using standardized mean importance values

To assess the importance of a morphological tag $m$ for predicting the entity tag $k$ across the corpus, we examine the $m$th dimension of $\hat{\mu}_k$. We chose this approach instead of assigning higher importance to features which are used to explain more instances throughout the corpus as in the original LIME approach [55]. In this way, we avoid falsely marking very common features as important. For instance, the morphological tag 'Case=Nom' which indicates the nominal case is commonly related to many words related to most entity tags. If we were to assign a high importance according to the number of occurrences across the corpus, we would incorrectly consider this type of features as important. Using standardized mean importance values in $\hat{\mu}_k$ is better in this regard.

We rank the morphological tags ($m$) for each entity tag $k$ using $\hat{\mu}_k(m)$. The ranked morphological tags for Finnish and Turkish can be seen in Tables 5.4 and 5.7, respectively. To conserve space, these tables show only the tags that appear in the top 8 (for Finnish) and top 20 (for Turkish). We also include the tables that contain all morphological tags in the appendix (Table A.1-A.8). The rows in these tables are the morphological tags ($m$) and the columns are the entity tags ($k$). Each cell gives the rank of the corresponding $\hat{\mu}_k(m)$. A high rank (1 being the highest) indicates a positive relation, whereas a low rank indicates a negative relation with respect to the prediction of entity tag $k$.

5.4.3.1. Finnish. The five morphological tags with the highest and lowest ranks in a column exhibit a coherent picture. The highest ones are generally related to the entity tag, while the lowest ones are either unrelated or are in contradiction with the semantics of the entity tag. For example, for Finnish, the first seven morphological tags in column 'LOC' of Table 5.4 include five case related tags which indicate the inessive 'Case=Ine', genitive 'Case=Gen', elative 'Case=Ela', illative 'Case=Ill', and adessive 'Case=Ade' cases. All of these cases are related to the locative semantics of the attached word. The column for 'TIM' also shows a similar relation. The essive case marker 'Case=Ess' which is related with temporal semantics is in the top three morphological markers of the Finnish 'TIM' entity. Additionally, we observe that the illative 'Case=Ill' and innessive 'Case=Ine' cases are among the most negative ones (87th and 89th) in the column of the 'DATE' entity tag. This is expected because these are known to be related to location expressions. Such observations are useful in assessing the quality of a trained model.

The significance of morphological tags is further examined by exploiting the rules of a rule-based NER tagger that was originally used to validate the Finnish dataset by other researchers [104, 118].

The Finnish dataset was curated with manual annotations. It was subsequently validated with the rule-based FiNER NER tagger to improve its quality [123]. FiNER is

Table 5.4. The ranks of $\hat{\mu}_k(m)$ for each entity tag in Finnish. The morphological tags that are in the first 8 for at least one entity tag are shown.

| Morphological Tag | ORG | TIT | PER | TIM | LOC | DATE | PRO | MISC | EVENT | OUTSIDE | median | std. dev. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Number=Sing | 89 | **1** | 89 | 89 | 89 | 88 | 89 | **2** | **3** | 89 | 89 | 42 |
| Case=Nom | 88 | 89 | **3** | 82 | 88 | 11 | **1** | 89 | **1** | **1** | 3 | 39 |
| Voice=Act | **3** | 83 | **5** | 80 | 79 | **6** | 12 | 83 | 84 | 85 | 12 | 30 |
| VerbForm=Fin | **6** | 79 | **9** | 61 | **8** | 12 | **8** | 78 | 12 | 86 | 8 | 26 |
| Mood=Ind | **8** | 81 | 70 | 52 | **9** | 14 | **9** | 81 | 16 | 88 | 9 | 27 |
| Number=Plur | **4** | 85 | **4** | 86 | **3** | 23 | 78 | **1** | 89 | 75 | 4 | 35 |
| Case=Gen | 87 | 87 | 88 | 79 | **2** | **7** | 88 | 13 | 87 | **2** | 87 | 37 |
| Degree=Pos | **2** | 88 | 85 | **2** | 87 | **1** | 82 | 84 | **6** | **3** | 82 | 41 |
| Person=3 | 20 | 84 | **7** | 54 | 17 | 33 | 76 | 66 | 28 | 83 | 20 | 19 |
| Case=Par | 17 | 86 | 87 | 85 | **6** | 86 | **2** | 85 | **7** | 14 | 7 | 28 |
| Style=Coll | 14 | 74 | **2** | **9** | 21 | 32 | 16 | 50 | 32 | 22 | 16 | 22 |
| VerbForm=Part | 23 | 65 | 57 | 76 | 39 | **3** | 33 | 73 | **8** | 36 | 33 | 27 |
| Case=Ine | 84 | 15 | 24 | **4** | **1** | 89 | **5** | 87 | 85 | 35 | 24 | 35 |
| Case=Ill | 33 | 14 | 77 | 75 | **5** | 87 | **3** | 82 | 31 | 34 | 31 | 29 |
| PartForm=Past | 25 | 69 | 60 | 77 | 74 | **5** | 29 | 65 | 10 | 33 | 29 | 26 |
| Case=Ela | 83 | 62 | 68 | 67 | **4** | 19 | **4** | 86 | 42 | 32 | 42 | 24 |
| VerbForm=Inf | 13 | 31 | 21 | 49 | 84 | 8 | 84 | 12 | 86 | 77 | 84 | 25 |
| Voice=Pass | 27 | 70 | 56 | 55 | 19 | **4** | 79 | 71 | 27 | 31 | 27 | 25 |
| Person=1 | **9** | **8** | 73 | 69 | 11 | 26 | **6** | 75 | 29 | 78 | 11 | 28 |
| Case=Ade | 86 | 37 | 69 | 66 | **7** | 30 | 11 | 80 | 82 | **4** | 69 | 26 |
| Connegative=Yes | 22 | 63 | 20 | **7** | 26 | 15 | 23 | **6** | 17 | 84 | 22 | 32 |
| Person[psor]=3 | 55 | 34 | 38 | 60 | 76 | 43 | 18 | **3** | 81 | 26 | 55 | 18 |
| Case=All | 54 | 75 | 81 | 65 | 12 | 36 | 13 | 79 | **2** | 23 | 13 | 22 |
| Derivation=Minen | 31 | **3** | 83 | 83 | 22 | 29 | 15 | **7** | 19 | 20 | 22 | 28 |
| NumType=Card | 16 | 68 | 30 | 88 | 82 | 21 | 87 | 77 | **5** | **6** | 30 | 32 |
| Case=Ess | 35 | 28 | 55 | **3** | 13 | **2** | 26 | 63 | 34 | **7** | 34 | 23 |
| Case=Tra | 53 | 12 | 17 | 51 | 70 | 28 | 43 | **9** | 38 | **8** | 43 | 16 |
| Typo=Yes | 28 | 76 | 64 | **5** | 83 | 24 | 40 | 64 | 13 | **9** | 40 | 29 |
| Mood=Imp | 34 | 23 | **8** | **8** | 25 | 17 | 75 | 16 | 23 | **5** | 25 | 6 |
| Abbr=Yes | **7** | **5** | **1** | 22 | 78 | 82 | 85 | 74 | **4** | 82 | 7 | 32 |
| Foreign=Yes | **5** | 11 | 86 | 27 | 23 | 69 | 17 | **5** | 11 | 81 | 17 | 30 |
| Clitic=Kin | 29 | 18 | **6** | **6** | 81 | 79 | **7** | **8** | 33 | 18 | 29 | 27 |
| Derivation=Lainen | 41 | **2** | 78 | 71 | 16 | 71 | 22 | 70 | 18 | 62 | 22 | 26 |
| NumType=Ord | 11 | **6** | 75 | 59 | 33 | 84 | 81 | **4** | **9** | 64 | 33 | 32 |
| Derivation=Inen | 79 | **4** | 71 | 62 | 27 | 34 | 73 | 19 | 21 | 69 | 71 | 24 |
| Derivation=Vs | 48 | **7** | 44 | 74 | 48 | 75 | 47 | 68 | 69 | 46 | 48 | 25 |
| *UNKNOWN* | **1** | 77 | 84 | **1** | 86 | 85 | 86 | 88 | 83 | 76 | 84 | 32 |

a part of the `finnish-tagtools` toolkit which contains a morphological analyzer, a tokenizer, a POS tagger, and a NER tagger for Finnish [123]. The rules of FiNER have been specified by linguists and tested on the dataset. As such, the labels produced by them may be considered as gold labels.

The FiNER authors define a rule for each named entity tag that matches every instance of it. To tag a sentence, it is tokenized and the morphological tags of every word are determined using a Finnish morphological analyzer. These morphological tags are then disambiguated using a POS tagger. The output of these tools consists of the surface form, the lemma, the disambiguated morphological tags, and some extra labels such as proper name indicators. The rules for each named entity tag are matched against this output in the order of rules definition file. The first successful match designates the named entity tag. Each rule is either a regular expression or a combination of other rules. Rules may be combined via a concatenation, union, or intersection using `pmatch` syntax [124]. For example, the rule named `PropGeoLocInt` in Table 5.5 matches a single word only if the word's morphological tag includes 'NUM=SG', and one of the three case morphemes 'CASE=INE', 'CASE=ILL', 'CASE=ELA', and the proper noun label 'PROP=GEO' (e.g. Finnish word 'Kiinassa' which means 'in China' matches this rule). In rows 2 and 3 of the table, the simple rules called 'Field' and 'FSep' are used to match a string of any length and a tab character, respectively. In rows 4 and 5 of the table, we see a specific concatenation of these two rules and several string literals in curly brackets. These pieces make up the rule that matches the surface form, the lemma, the disambiguated morphological tags, and the extra labels. However, a successful match of this rule does not result in a named entity tag. Instead, it is used in more general rules as seen in the definition of `LocGeneral2` in rows 7-8. It matches the rule `PropGeoGen` and the rule `PropGeoLocInt` in the right context. This hierarchical structure continues up to the top rule for the 'Location' named entity tag, namely the `Location` rule. The morphological tags used in FiNER are different from the ones in our paper as it employs the `Omorfi` morphological analyzer. However, the mapping is straightforward except for a few cases.

To examine the correspondence between the results produced by our method and

FiNER rules, we construct a graph. The internal nodes in the graph correspond to the rule names and the leaf nodes to the morphological tags. We define an edge from rule A to rule B if and only if the definition of rule A contains a reference to rule B in the form of concatenation, union or intersection operators. We process this graph to produce a subgraph for each of the nodes that correspond to named entity tags. For this, we start from the node of a named entity tag and traverse the graph in breadth-first fashion. The resulting subgraph is not a tree, however it is highly hierarchical. Figure 5.7 shows a subgraph for 'Location', which demonstrates the hierarchical nature of the FiNER rules in the graph. In the figure, 'PropGeoGen' is referred to by two rules ('LocGeneral2' and 'LocGeneralColloc1'), which themselves are referred to by 'LocGeneral'. In a subgraph, we follow every possible path from the root node to the leaf nodes that are composed of the `Omorfi` equivalents of the morphological tags from Table 5.4. If there is at least one such path, we assume that the morphological tag is related to this named entity tag and thus may be used to evaluate the list of important tags produced by our method.

Table 5.6 shows the results of the correspondence examination between the proposed explanation method and the FiNER tagger for the five most-frequently occurring (more than 1,000 occurrences) named entity tags in the dataset. We ignored the morphological tags that are not in the top 10 of the importance list of the proposed explanation method. This resulted in a set of 19 morphological tags which was used during the following evaluation. We quantify the rate of correspondence by counting the number of successful matches between our method and the FiNER rules. When it is concurrently true that a morphological tag is predicted as important for a named entity tag and there is at least one path from the named entity to the morphological tag, we regard this as a true positive (TP). If it is predicted to be important by our method but no paths exist between the named entity tag and the morphological tag, it is counted as a false positive (FP). All predictions for 'Location' are correct, i.e. all of our predicted morphological tags are reachable from the named entity tag. However, five morphological tags that are predicted as unimportant to 'Location' have paths originating from the 'Location' named entity tag. These are regarded as false negative (FN) predictions. The remaining seven predictions are counted as true negatives (TN).

In Table 5.6, we see a mostly assuring picture. The precision rates of 'Location' and 'Organization' are quite high while it is lower for 'Person'. This is due to the fact that two of the three false positives in 'Person' ('Style=Coll' and 'Person=3') are absent in the FiNER rules. 'Style=Coll' is a tag specific to this dataset and 'Person=3' does not exist in FiNER rules although it is a tag found in our corpus. The single false positive example in 'Organization' is also due to the 'Person=3' tag. The worst recall rate occurs with our predictions for the 'Product' named entity tag. The recall ratio indicates that we miss about 75% of the morphological features which are important according to the FiNER rules. Some of the missed ones are among the most common morphological tags such as 'Case=Gen' and 'Number=Plur', which are used in many basic rules such as 'PropGeoGen'. These basic rules appear in many paths that start from any named entity tag. This inevitably results in the existence of many paths to these morphological tags for each named entity tag, thus lowering the recall rate for all. This observation is also valid for all other named entity tags in the sense that these common tags are included within their false negatives.

5.4.3.2. Turkish. An inspection of the 'ORG', 'LOC', and 'PER' entity tag columns in Table 5.7 for Turkish reveals that the tag that indicates proper nouns ('Prop') is the dominant one. This shows that the model relies on the morphological analyzer's performance to mark proper nouns correctly. However, the case of 'P3sg' is more interesting. This morpheme is commonly found in noun clauses which are organization or location names. On the other hand, it is never attached to person names. The case of 'P3pl' is similar. This is reflected in our results; these morphological tags are not positively related with the 'PER' entity tag as seen in the table. The case of 'Almost' is interesting as it is a rare morpheme and is almost never attached to the correct morphological analysis. These properties should have made it an unimportant tag. On the contrary, it is regarded as an important tag for 'ORG' and 'LOC' named entity tags by our method. One possible explanation is that when 'Almost' is removed from the feature sets to create perturbed sentences, the morphological analyses that contained 'Almost' prior to the perturbation is regarded more probable by the tagger, which in turn decreases the probability of the prediction. This causes the morpheme 'Almost'

to be considered as an important morphological tag with explanatory value.

### 5.4.4. Importance distributions of morphological tags

The histograms of importance values for various combinations of entity types and morphological tags are shown in Figure 5.8 for the Finnish NER corpus. Each row in Figure 5.8a is a heatmap that represents a histogram of values in $\hat{\mathbb{E}}^{\texttt{PER}}(m)$, which is a vector that consists of all importance values of the morphological tag $m$ explaining a 'PER' entity tag prediction in a region. Bin edge positions are determined by $-10^{i/10}$ and $10^{i/10}$ for the negative and positive sides, respectively, where $i \in \{-25, \ldots, 13\}$. The frequency corresponding to each bin is coded with color tones from white to black. The morphological tags are presented in descending order according to their mutual information gain $MI_{\texttt{PER},m}$. Only the first 20 morphological tags are shown here due to space constraints. Figure 5.8b is formed likewise. When all morphological tags are considered instead of these, we observe that there is a clustering between $-0.050$ and 0.063. However, this prominent cluster fades away and new clusters emerge when we focus on the higher ranked morphological tags. We argue that this is correlated with high mutual information gain values corresponding to higher ranked morphological tags.

Figures 5.8c and 5.8d show that the explanation values for the morphological tags 'Case=Ine' and 'Case=Nom' are distributed in a different way by plotting the histograms of $\hat{\mathbb{E}}^{\texttt{LOC}}(m)$ where $m$ is the corresponding dimension. These figures have the same $x$ axis as in Figures 5.8a and 5.8b. We should note that the $x$ axis is in log-space so that the clusters near the center are very close to zero, whereas the concentration around 7.94 indicates that a significant portion of the importance values are high.

### 5.4.5. Importance of morphological tags across the entity tags

To determine the morphological tags that are important across entity tags, we count the number of times the rank of $\hat{\mu}_k(m)$ is in the top or bottom 10 ranks. The morphological tags are sorted by the sum of these frequencies for the features that

are ranked at the top and bottom of the list. The first 10 morphological tags with the highest sum for Finnish are shown in Table 5.8 which are the most frequently encountered tags for most languages. They signify singular or plural, active or passive, and mark the word as nominal, genitive, or inessive cases.

### 5.4.6. Quantitative validation using mutual information

In order to validate the explanations created by the proposed explanation model, we employ $\hat{\mu}_k$ (Equation 5.8d) and $MI_{k,m}$ (Equation 5.10). We denote the 10 morphological tags ($m$) with the highest $\hat{\mu}_k(m)$ values as $\mathbb{I}_k$. Independently, we calculate the mutual information gain $MI_{k,m}$ between the probability of each morphological tag $m$ being in region $r_{ij}$ and the probability of entity tag $k$ being the label of region $r_{ij}$. We call the first 10 morphological tags with the highest mutual information score as $\mathbb{J}_k$.

$\mathbb{I}_k$ represents the proposed method's list of globally important morphological tags, whereas $\mathbb{J}_k$ is a list created by information gain independent of any particular model. The degree of agreement between these lists gives a quantifiable metric to evaluate different explanation methods. We proceed to take the intersection of $\mathbb{I}_k$ and $\mathbb{J}_k$ for each entity tag $k$ and report the common morphological tags in Table 5.9 for Finnish and Turkish. The number of morphological tags that are both in $\mathbb{I}_k$ and $\mathbb{J}_k$ hints that the proposed explanation method can correctly predict the morphological tags with high information gain.

### 5.4.7. The impact of the absence of a morphological tag

After the rankings of the morphological tags using the average importance values $\hat{\mu}_k(m)$ are obtained for each morphological tag $m$ and entity tag $k$ (Tables 5.4 and 5.7), we consider how this information could be used to improve our model. One idea was to modify the architecture of the NER tagger so that it pays more attention to the higher ranked tags compared to the other tags. For example, an extra dimension in the morphological tag embedding to represent the rank of the corresponding morphological tag can be exploited by the neural network. However, as the results of the explanation

method are relevant only in the context of the specific model that is being inspected, this would result in a new model related to the original model. If the original model was successful in exploiting the morphological features that are really important to the NER task, this approach would yield successful. On the other hand, if the original model was not able to exploit the important morphological tags due to the training regime or the inefficiency of the architecture, our method would falsely indicate other morphological tags instead of the important tags. This approach would yield a model with reduced performance. So, training a model which exploits the higher ranked morphological tags reported in our study might not result in an improved performance.

Instead, we decided to test the hypothesis that higher ranked morphological tags may improve the performance for NER by following a corpus-based approach. For the 'Location' named entity tag in Finnish, we chose the top two ranked morphological tags ('related tags') and eight other randomly selected morphological tags which are not ranked within the first or last 10 positions ('unrelated tags'). For each of these 10 tags, we created a new version of the dataset so that no morphological analysis contains the corresponding tag. We then trained and evaluated each of the 10 models separately in two independent runs. We calculated the averages of the F-measure, precision, and recall metrics for the 'Location' named entity tag using these two runs. Table 5.10 compares the performance of the 'related tags' ('Case=Ine' and 'Case=Gen') and the 'unrelated tags' in terms of the differences in these metrics. For each combination of two 'related' and eight 'unrelated' tags, we subtract the success rate of the model in which a related tag is removed from the success rate of the model in which an unrelated tag is removed. The average, minimum, and maximum values of the resulting eight difference values are shown in the respective columns. A positive difference in the average column indicates that the removal of the unrelated tags decreases the performance of the model less than the removal of the related tag, while a negative difference indicates the opposite.

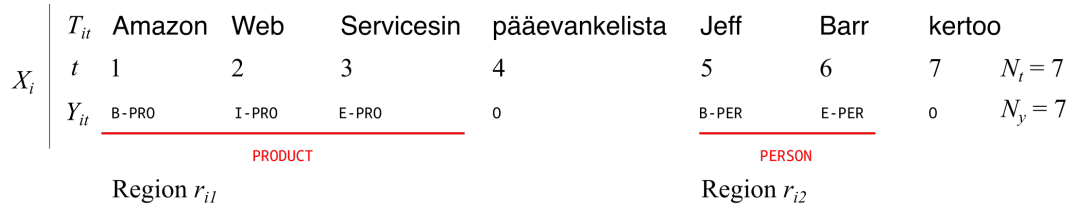
The results shown in Table 5.10 are contrary to our expectations. Our hypothesis that the 'related tags' contain a stronger signal for the named entity tag and their absence would decrease the model performance is not verified. This is verified only for

the precision metric of the 'Case=Ine' tag and the recall metric of the 'Case=Gen' tag. An observation might explain the failure to reject the null hypothesis. An inspection of the morphological analyses reveals that 'unrelated tags' occur less frequently than 'related tags'. We have observed that there are some morphological tags that co-occur with each morphological tag and the removal of one of them might be compensated by the co-occurring tags. However, this mechanism may not hold true for 'unrelated tags' as they have relatively fewer co-occurring morphological tags. This might in turn result in a higher loss of performance when an 'unrelated tag' is removed.

## 5.5. Conclusions

In this chapter, we introduced an explanation method which can be employed for any sequence-based NLP task. We introduce a procedure which can be adopted to any model that implements a sequence-based NLP task by transforming the model into a multi-class classification model. A case study that uses a joint NER and MD tagger is presented that demonstrates that the proposed method can be employed to provide explanations for single input samples to assess the contribution of features to the prediction. Furthermore, it is shown that an analysis of these explanations across the corpus can be helpful in assessing the plausibility of a given trained model.

While forming explanations, we treat each feature in a given region as independent from each other. However, features may be related to each other in many ways. Firstly, some morphological tags in a single morphological analysis of a given word are dependent on each other. For instance, the presence of one tag may strongly signal the presence of another tag or the order of appearance in the morpheme sequence may be important. Secondly, this dependence may be observed between features inside and outside the region. For example, named entities are usually related to the features of the words to their left and right, such as the morphological tags and the characters of the surface forms. In future work, we aim to consider such relations by extending our model to permit perturbation across multiple regions of variable sizes. The dependency between features can be explored better if our method allows perturbing one or more of the features to the left or the right of the region associated with a named entity.




| $t$ | Morphological analyses | Feature sets |
|---|---|---|
| 1 | - Amazo\|~NOUN~N~Case=Gen\|Number=Sing<br>- Amaz\|~NOUN~N~Case=Ill\|Number=Sing<br>- Amazo\|~PROPN~N~Case=Gen\|Number=Sing<br>- Amazon\|~PROPN~N~Case=Nom\|Number=Sing<br>- Amazon\|~ADJ~A~Case=Nom\|Degree=Pos\|Number=Sing<br>- Amazko\|~NOUN~N~Case=Gen\|Number=Sing<br>- Amazon\|~PROPN~N~<br>- Amazto\|~NOUN~N~Case=Gen\|Number=Sing<br>- amazon\|~NOUN~N~Case=Nom\|Number=Sing | $F_{i1}$<br>Case=Gen<br>Number=Sing<br>Case=Ill<br>Case=Nom<br>Degree=Pos |
| 2 | - Web\|~PROPN~N~Case=Nom\|Number=Sing<br>- Web\|~PROPN~N~<br>- Web\|~PROPN~N~Case=Nom\|Number=Sing<br>- Web\|~NOUN~N~Abbr=Yes\|Case=Nom\|Number=Sing<br>- web\|~NOUN~N~Case=Nom\|Number=Sing | $F_{i2}$<br>Case=Nom<br>Number=Sing<br>Abbr=Yes<br>Case=Nom |
| 3 | - Services\|~PROPN~N~Case=Gen\|Number=Sing<br>- Servicesi\|~PROPN~N~Case=Gen\|Number=Sing<br>- Servicesin\|~ADV~Adv~<br>- Servicess\|~PROPN~N~Case=Gen\|Number=Sing\|Typo=Yes<br>- Servicest\u00e4\|~VERB~V~Mood=Ind\|Number=Sing\|Person=1\|Tense=Past\|VerbForm=Fin\|Voice=Act<br>- Servicet\u00e4\|~VERB~V~Mood=Ind\|Number=Sing\|Person=1\|Tense=Past\|VerbForm=Fin\|Voice=Act<br>- Servicet\u00e4\u00e4\|~VERB~V~Mood=Ind\|Number=Sing\|Person=1\|Tense=Past\|VerbForm=Fin\|Voice=Act | $F_{i3}$<br>Case=Gen<br>Number=Sing<br>Typo=Yes<br>Mood=Ind<br>Person=1<br>Tense=Past<br>VerbForm=Fin<br>Voice=Act |

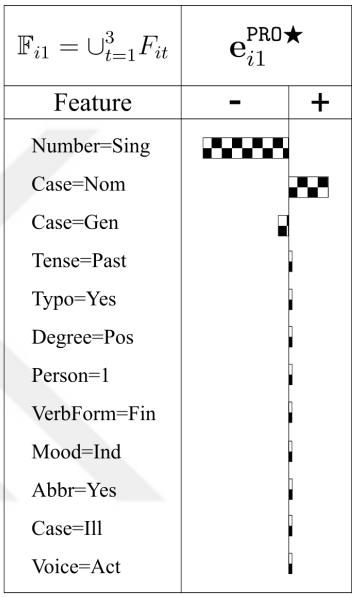



Figure 5.3. Explanation of a NER tagger's prediction where $N_t = 7, N_f = 7, N_y = 7, N_o = 0$. $X_i$ is the $i$th sentence. Only first seven tokens of the sentence are shown. They translate to "Jeff Barr, the chief evangelist of Amazon Web Services, says ...". Region $r_{i1}$ is labeled as a 'PRO' entity tag. All possible morphological analyses of first, second, and third tokens ($t$) in region $r_{i1}$ that are used to form feature sets $F_{it}$ are shown. $\mathbb{F}_{i1}$ is the set of morphological tags resulting from the union of all $F_{it}$ in region $r_{i1}$. $\mathbf{e}_{i1}^{\text{PRO}\bigstar}$ is the resulting explanation vector.

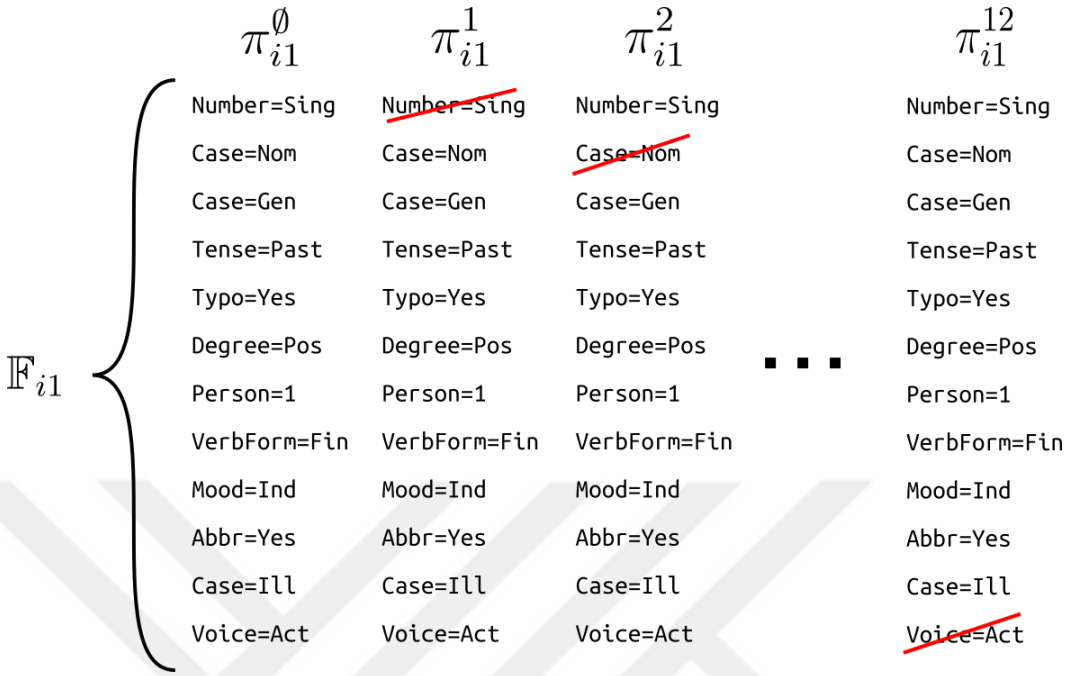



Figure 5.4. The change in $\mathbb{F}_{i1}$ corresponding to every perturbed sentence $\pi^{r}_{i1}$ is depicted. Crossed out morphological tags are removed from $\mathbb{F}_{i1}$. Morphological tags which are not already present are not shown.

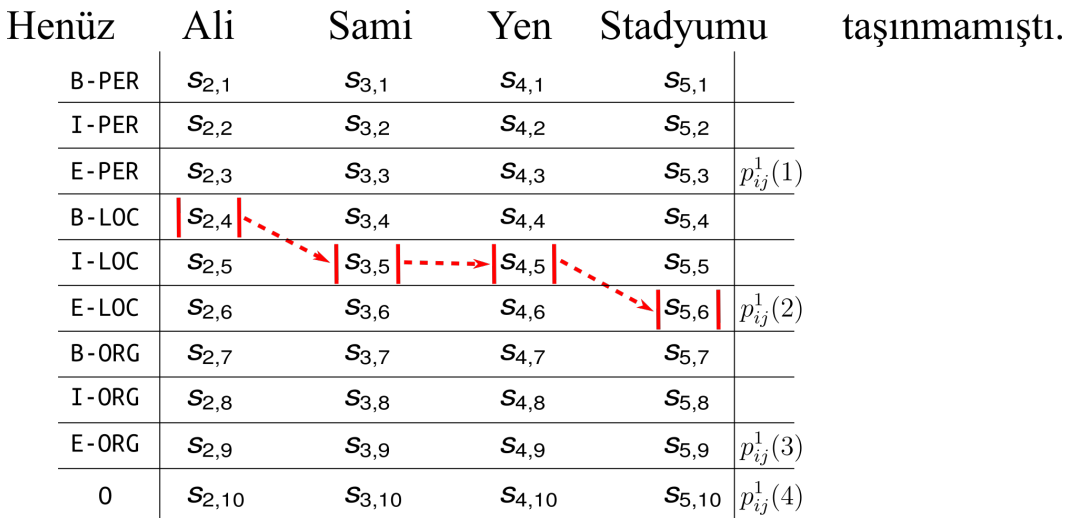



Figure 5.5. A NER task for a Turkish sentence (meaning "Ali Sami Yen Stadium had not yet been relocated."). The token-level tag predictions for each position are shown with scores which are denoted with $s_{t,o}$. The correct sequence of the token-level named entity tags are marked in red. Single token entities are not applicable to multiple token entities, thus their scores are N/A.

$X \leftarrow$ set of sentences to be explained

$K \leftarrow$ number of classes

**for** $i = 1$ to $|X|$ **do**

  **for all** region $j$ in sentence $X_i$ **do**

    $\mathbb{F}_{ij} \leftarrow$ set of features in $r_{ij}$

    $\pi_{ij} = \{\texttt{remove}(X_i, j, f) : f \in \mathbb{F}_{ij}\} \cup \{X_i\}$

    **for all** perturbed sentence $\pi_{ij}^r$ in $\pi_{ij}$ **do**

      $p_{ij}^r \leftarrow$ vector of probabilities of all labels using $\texttt{predict}(\pi_{ij}^r)$

      $\Delta\mathbb{P}_{ij} \leftarrow$ filled such that $r$th column is $p_{ij}^r - p_{ij}^{\emptyset}$ and the last column is $\vec{0}$

      $\mathbb{C}_{ij}(r, :) \leftarrow$ vector of zeros, minus ones, and ones representing the perturbed sentence $\pi_{ij}^r$

    **for all** label $k$ **do**

      $\mathbf{e}_{ij}^{k\bigstar} = \text{argmin}||\Delta\mathbb{P}_{ij}(k, :) - \mathbb{C}_{ij}e_{ij}^k||_2^2 + ||e_{ij}^k||_2^2$

Figure 5.6. The explanation method for sequence-based NLP tasks. `predict` function relies on the model to obtain the probabilities for each label $k$.

Table 5.5. Three FiNER rules are presented in the order of increasing generality. The first one acts on single words. The second applies a rule on a single word and requires the right context to match another rule. The last one is the top rule for the 'Location' tag. Options 1, 2, 3, and 6 can also be seen in Figure 5.7 as they lie on paths that reach a morphological tag while other options do not lead to any.

| | | |
|---|---|---|
| 1 | **Rule** `PropGeoLocInt` | |
| 2 | **Surface form** | Field FSep —i.e. any string is allowed |
| 3 | **Lemma** | Field FSep —i.e. any string is allowed |
| 4 | **Morphological tags** | Field [{NUM=SG} Field {CASE=}[{INE} \| {ILL} \| {ELA}]] Field FSep |
| 5 | **Extra labels** | Field [{PROP=GEO}] Field FSep |
| 6 | **Rule** `LocGeneral2` | |
| 7 | **Single word** | PropGeoGen |
| 8 | **Right context** | RC( WSep PropGeoLocInt ) |
| 9 | **Rule** `Location` | |
| 10 | **Option 1** | Ins(LocGeneral) |
| 11 | **Option 2** | Ins(LocGeogr) |
| 12 | **Option 3** | Ins(LocPolit) |
| 13 | **Option 4** | Ins(LocStreet) |
| 14 | **Option 5** | Ins(LocAstro) |
| 15 | **Option 6** | Ins(LocPlace) |
| 16 | **Option 7** | Ins(LocFictional) |

Table 5.6. Results of the matching between the proposed explanation method and the FiNER tagger for the five most-frequently occurring named entity tags.

| | TP | TN | FP | FN | Precision % | Recall % | F1-measure % |
|---|---|---|---|---|---|---|---|
| Location | 7 | 7 | 0 | 5 | 100.00 | 58.33 | 73.68 |
| Organization | 4 | 8 | 1 | 6 | 80.00 | 40.00 | 53.33 |
| Person | 4 | 8 | 3 | 4 | 57.14 | 50.00 | 53.33 |
| Product | 2 | 7 | 4 | 6 | 33.33 | 25.00 | 28.57 |
| Time | 1 | 9 | 7 | 2 | 12.50 | 33.33 | 18.18 |
| Total | 18 | 49 | 22 | 25 | 45.00 | 41.86 | 43.37 |

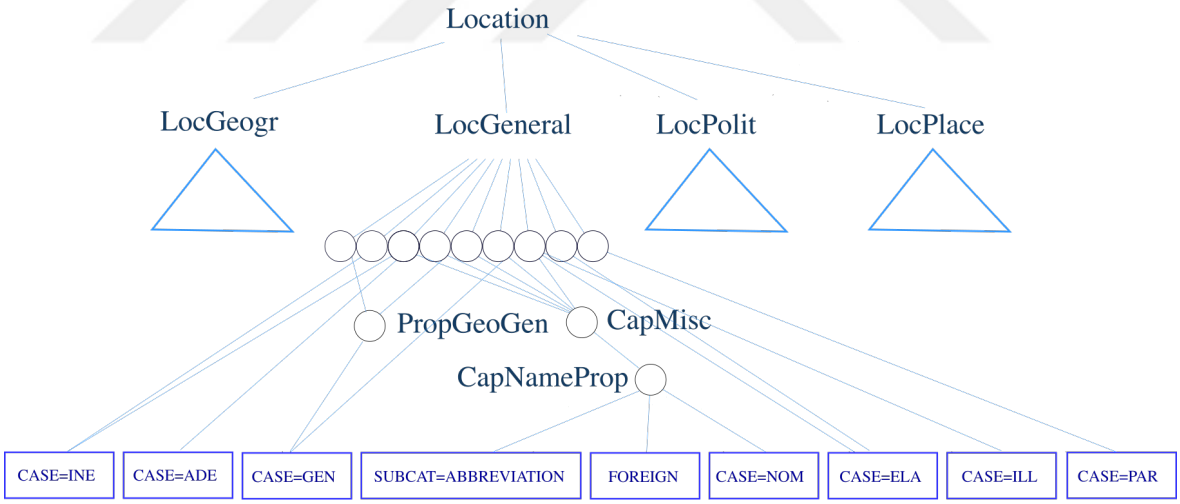
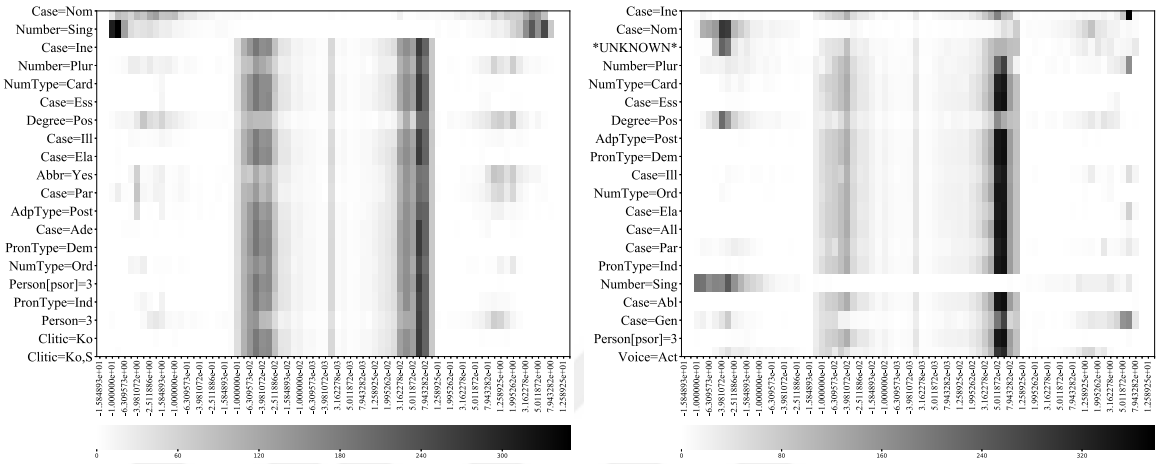



Figure 5.7. A subset of the graph that is formed based on the FiNER rules. It shows the nodes and the connections that lie between the 'Location' top rule and the morphological tags reachable using the rules. We only show the subgraph reachable from 'LocGeneral' in finer detail, while replacing the other subgraphs with triangles. Morphological tag nodes are represented with a rectangle. Circles represent the internal rules.

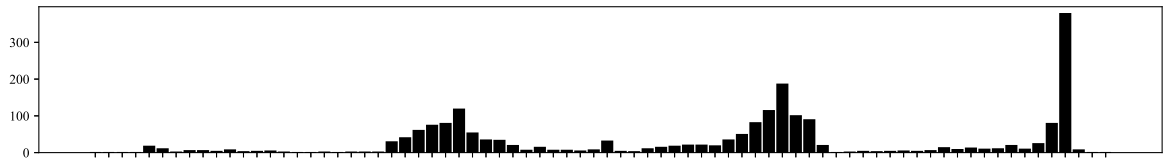
Table 5.7. The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown.

| Morphological Tag | ORG | LOC | PER | median | std. dev. |
|---|---|---|---|---|---|
| Noun | 176 | 181 | **2** | 89 | 0 |
| A3sg | **2** | 180 | 178 | 90 | 0 |
| Verb | **8** | 147 | 173 | 90 | 0 |
| Adj | **3** | 170 | 176 | 89 | 0 |
| P3sg | **4** | **2** | 169 | 86 | 0 |
| Posˆ DB | **17** | 160 | 171 | 94 | 0 |
| Punc | **20** | **5** | **7** | 13 | 0 |
| Pos | **13** | **16** | **4** | 8 | 0 |
| Prop | **1** | **1** | **1** | 1 | 0 |
| Acc | **12** | 169 | 174 | 93 | 0 |
| Nomˆ DB | **9** | 166 | 170 | 89 | 0 |
| Loc | **19** | 177 | 161 | 90 | 0 |
| Adverb | 46 | **14** | **19** | 32 | 0 |
| Conj | 28 | **6** | **18** | 23 | 0 |
| Num | 21 | **13** | **9** | 15 | 0 |
| Pass | 59 | **19** | **17** | 38 | 0 |
| P3pl | **14** | **12** | 70 | 42 | 0 |
| Adjˆ DB | **15** | 172 | 156 | 85 | 0 |
| Past | 62 | 153 | **14** | 38 | 0 |
| Card | 36 | **18** | **16** | 26 | 0 |
| Imp | **16** | **7** | **6** | 11 | 0 |
| A2sg | **18** | **8** | **5** | 11 | 0 |
| Ness | **11** | 150 | 147 | 79 | 0 |
| Aorˆ DB | 32 | **17** | 164 | 98 | 0 |
| Agt | 22 | **9** | 172 | 97 | 0 |
| With | 27 | 158 | **10** | 18 | 0 |
| P1sg | 24 | 30 | **8** | 16 | 0 |
| Almost | **6** | **4** | 139 | 72 | 0 |
| A2pl | 41 | 28 | **11** | 26 | 0 |
| Opt | 51 | 29 | **13** | 32 | 0 |
| Nounˆ DB | **5** | **3** | 138 | 71 | 0 |
| Equ | 37 | **15** | **15** | 26 | 0 |
| Ord | 26 | **11** | **12** | 19 | 0 |
| *UNKNOWN* | **10** | 165 | **3** | 6 | 0 |
| Pron | 56 | 55 | **20** | 38 | 0 |
| Interj | 173 | **20** | 32 | 102 | 0 |

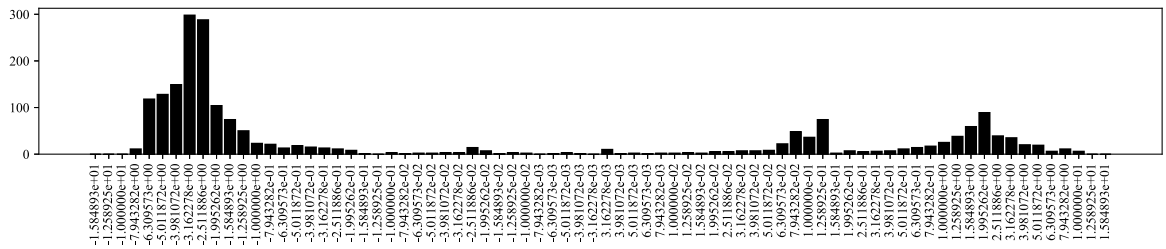



(a) Each row is a heatmap that represents a histogram of $\hat{\mathbb{E}}^{\texttt{PER}}(m)$ corresponding to the morphological tag $m$.

(b) Each row is a heatmap that represents a histogram of $\hat{\mathbb{E}}^{\texttt{LOC}}(m)$ corresponding to the morphological tag $m$.




(c) The histogram of $\hat{\mathbb{E}}^{\texttt{LOC}}(\texttt{Case} = \texttt{Ine})$ which is the first row of (b).




(d) The histogram of $\hat{\mathbb{E}}^{\texttt{LOC}}(\texttt{Case} = \texttt{Nom})$ which is the second row of (b).

Figure 5.8. The histograms of importance values for various entity tag and morphological tag combinations. Darker colors indicate higher frequencies. Calculated over the training corpus in Finnish. All histograms in the figures share the same bin edges. Edge positions are calculated by $-10^{i/10}$ and $10^{i/10}$ for the negative and positive sides, respectively, where $i \in \{-25, \ldots, 13\}$.

Table 5.8. The frequency of ranks that are in the first ten of an entity tag for each morphological tag in Finnish.

| Morphological Tag | Top | Bottom | Top+Bottom |
|---|---|---|---|
| Number=Sing | 3 | 7 | 10 |
| Degree=Pos | 5 | 5 | 10 |
| Case=Nom | 4 | 5 | 9 |
| Case=Gen | 3 | 5 | 8 |
| Case=Par | 3 | 5 | 8 |
| *UNKNOWN* | 2 | 6 | 8 |
| Number=Plur | 4 | 3 | 7 |
| Voice=Act | 3 | 4 | 7 |
| Abbr=Yes | 4 | 3 | 7 |
| Case=Ine | 3 | 4 | 7 |

Table 5.9. Common Finnish and Turkish morphological tags that are both in $\mathbb{I}_k$ and $\mathbb{J}_k$.

| Finnish | | |
|---|---|---|
| Entity tag | Morphological tags | Agreement rate |
| ORG | 'Case=Gen', 'Number=Plur', 'Number=Sing', 'Case=Nom', 'Degree=Pos', '*UNKNOWN*' | 0.6 |
| TIT | 'Case=Gen', 'Case=Par', 'Number=Sing', 'Case=Nom' | 0.4 |
| PER | 'Number=Plur', 'Number=Sing', 'Case=Nom', 'Abbr=Yes', 'Degree=Pos' | 0.5 |
| TIM | 'Number=Sing', 'Case=Ess', 'Degree=Pos', 'Case=Par', 'PronType=Dem', '*UNKNOWN*' | 0.6 |
| LOC | 'Number=Plur', 'Case=Nom', 'Degree=Pos', 'Case=Ill', 'Case=Ine', '*UNKNOWN*' | 0.6 |
| DATE | 'Case=Ess', 'Case=Ine', 'Degree=Pos', '*UNKNOWN*' | 0.4 |
| PRO | 'Case=Par', 'Case=Nom', 'Case=Ela', 'NumType=Card' | 0.4 |
| MISC | 'Number=Plur', 'Number=Sing', 'Case=Ela', 'Person[psor]=3', 'Degree=Pos', 'Case=Par', '*UNKNOWN*' | 0.7 |
| EVENT | 'Case=Ine', ' Number=Plur', 'Voice=Act' | 0.3 |
| OUTSIDE | 'Case=Ade', ' Number=Sing', 'Case=Nom', 'Degree=Pos' | 0.4 |
| Turkish | | |
| Entity tag | Morphological tags | Agreement rate |
| ORG | 'Adj', 'P3sg', 'Prop', 'A3pl', 'Nom' | 0.5 |
| LOC | 'Loc', 'Dat', 'P3sg', 'Prop', 'Nom', 'Gen' | 0.6 |
| PER | 'Adj', 'Gen', 'Dat', 'Nom' | 0.4 |

Table 5.10. Comparison between each 'related tag' and 'unrelated tag' sets using F-measure and Recall metrics.

| | | **Unrelated tags** | | |
|---|---|---|---|---|
| **Related tag** | **Metric** | **Average % Difference** | **Min % Difference** | **Max % Difference** |
| **Case=Ine** | Precision | 0.31 | -1.83 | 2.86 |
| | Recall | -2.73 | -5.44 | -0.39 |
| | F-measure | -1.31 | -3.07 | -0.21 |
| **Case=Gen** | Precision | -2.34 | -4.48 | 0.21 |
| | Recall | 0.14 | -2.57 | 2.48 |
| | F-measure | -1.07 | -2.83 | 0.03 |

# 6. DISCUSSION AND FUTURE WORK

## 6.1. NER for Morphologically Rich Languages

As introduced in Chapter 3 and Chapter 4, we employ machine learning models that utilize neural networks to perform named entity recognition. These neural networks are designed so that their model capacities can be adjusted by setting the number of dimensions of several variables. For example, increasing the number of dimensions ($p$) of the cell variable of Bi-LSTM module may have a significant positive impact. However, this also causes the number of parameters to grow exponentially, e.g. $U^{(i)}$ has $p \times p$ parameters in Equation 3.1. Moreover, the number of experiments required for the ablation study is very high since we repeat each experiment 10 times to obtain better estimates for the F1-measure. For this reason, we tuned the number of dimensions of the cell variable and the word embeddings to a number that permitted the computations within the capacity of our computational resources. We took this decision hoping that the loss in model capacity would not negatively impact the results of the ablation study. It might be the fact that some parts of the model, such as morphological disambiguation, can only be effective when the cell dimension is higher than a certain threshold. If this value is greater than 10, then the results of our ablation studies would be questionable. This threshold should be investigated further to assess its impact. Additional computational resources are required to perform experiments with higher dimensions.

As stated in Section 3.3.1 and 4.1.1.1, we represent words using fixed-length vectors. We either learn the optimized parameters during training, or start with word embeddings that are trained using specific algorithms. We experimented with two algorithms: Word2Vec [34] and fastText [99]. These algorithms represent a word using its distributional statistics. They hypothesize that it is possible to guess the most probable word if the words in its proximity are known, i.e. the word in the middle of five consequent words in a moderate size sentence can be predicted in a significant number of cases. Based on this distributional behaviour, they determine the optimum

values for each word representation. After training, models like ours treat the fixed-length vectors associated with the words as representations of the words in their input sentence. However, these representations are invariant to their context (the other words in the sentence). For example, the word representation for the word 'can' in English is the same for both of the sentences: 'I can climb a tree' and 'This soda can is made of aluminium.' This causes a loss in the representation capacity of these algorithms. In recent years, several approaches that provide word representations that are dependent on the other words present in a given sentence were introduced to address this problem [125, 126]. These contextual word embeddings greatly enhanced the performance on many NLP tasks, which suggests that a similar performance improvement is possible in Turkish. Unfortunately, during our thesis work, we did not have a chance to experiment with these new algorithms for two reasons: i) we did not have the resources to compute these new representations ourselves ii) maximizing the F1-measure is not the main aim of our work. During this time, a study published contextual word embeddings for Turkish, which used these embeddings to predict NER tags and achieved a F1-measure of 95.55% [105]. Future work may experiment with these embeddings and test whether morphological embeddings would further improve the performance when using contextual word embeddings. This would be an important observation as it is thought that contextual word embeddings might solve the problem of representing words largely by capturing the semantic properties of words, but it is not known whether they are capturing morphological information in a form helpful for improving NER performance.

The model in Chapter 3 requires a morphological disambiguator to tag unseen sentences as it utilizes the disambiguated morphological tag for each word. This requirement is difficult to satisfy for some languages, since they are often not readily available. Moreover, even if a disambiguator exists, it is usually difficult to integrate into the pipeline of the system due to the restrictive architectural decisions made by the developer of the disambiguator. This problem is solved partially in Chapter 4 in which we introduce a NER tagger that jointly learns to disambiguate the morphological tags of each word. However, this model still requires a morphological analyzer to obtain all possible morphological analyses of each word. Future work may integrate a morphological analyzer to the model in addition to the morphological disambiguator.

This would eliminate the need for external components, rendering a system that is suitable for any language given that a pair of named entity recognition and morphological disambiguation datasets exists.

In the evaluation phase, we have employed the NER datasets that are frequently used in the literature [9, 13, 83, 102–104]. They serve as important resources when performing comparisons. It is useful to compare alternative methods against the same datasets to assess their relative performances. However, at the very best, they are a collection of sentences that are sampled uniformly from online news outlets or other types of written text. This uniformity may prevent new methods from standing out, as these datasets may not contain enough difficult-to-predict samples which only models that exploit more complex signals could effectively utilize. We know that the difficulty of predicting a named entity depends on both the words that make up the entity and the given sentence. For example, given the sentence "Real earned two on Friday", 'Real' may seem as mentioning a person with last name 'Real'. However, if the entity was composed of two words, i.e. "Real Madrid earned two on Friday", most people would recognize 'Real Madrid' as an organization entity. On the other hand, if we changed the first version a little bit, i.e. "Real earned two on Friday's match", it would be possible to guess that 'Real' refers to an organization entity. This kind of sentences are rare in current datasets. If there were datasets that include more challenging sentences, the methods that employ techniques that discriminate ambiguities better would perform significantly better than others. Unfortunately, as this is not the case, the state-of-the-art results in the literature is not improving much compared to other task challenges in other areas such as computer vision. For example, through 2011 and 2019, we observe only a 3% improvement in the state-of-the-art performance for NER. This is low when compared with the 75% improvement in ImageNet state-of-the-art performance. When we compare the state-of-the-art performances of 2016 and 2019, ImageNet's state-of-the-art improved 10% while NER's performance improved 2.5%, which means the improvement in ImageNet's state-of-the-art performance is only 4 times high. This is an improvement compared to the performance difference of 25 times of the former time frame. However, this also suggests that the impact of utilizing obscure signals might be playing a role instead of methods that improve the

generalization power. For this reason, further work on building a new type of NER dataset to be used in evaluation would be an important contribution.

Most NER taggers handle the problem in the context of a single sentence. However, entities are often understood based on their surrounding sentences. This thesis addresses NER in the conventional approach of dealing with a single sentence. It was not possible to follow the other path as most NER datasets for MRLs do not retain the order of the sentences or any other similar information that hint for contextual proximity. A useful future work would be to build such datasets for MRLs to extend existing research into multi-level contexts.

### 6.2. Explaining the predictions of NLP systems

The approach of our explanation method in Chapter 5 can be summarized as incrementally removing each feature to observe the difference it creates on the output of the model, then using this information to determine the impact of the features on the output labels. We focus only on morphological features related to the regions of the sentence where named entities occur. This introduces two concerns. First, some features may be interdependent, i.e. the presence of one feature may signal the presence or absence of another feature. This may interfere with the main mechanism of our explanation method because the examined model may exploit the clue given by the feature that is not removed. Thus the removal of a feature may not affect the prediction probability of a specific label. An example would be to remove the feature that signifies the noun part-of-speech tag while not removing the features that represent the genitive or dative cases since only nouns can be marked genitive or dative cases. Instead, the explanation method should assess the effect of a group of features by removing several features at once. However, this would introduce a feasibility problem due to the high number of possible groups among the features. Future work should explore methods to efficiently search the space of possible group of features. Instead of an exhaustive search, one could begin with a single member and employ some heuristic measures with a stopping criteria to form a group of features. These measures may be feature-feature and feature-label correlations, or custom metrics such as the change in the confidence

of the model for a specific label.

In this thesis, we have only considered removing features related to the region of a named entity. However, many named entity labels are not only dependent on the region of the named entity. This is apparent from the performance increase provided by using CRF models to predict the labels instead of models that employ token-level prediction. Also, using bidirectional LSTMs or RNNs improve the performance compared to using unidirectional sequential models to represent the context. Future work should also consider features from outside of the region of the named entity to capture such dependencies which could shed light on the effect of features to the specific label that we are trying to explain. This could be done while also considering groups of features as discussed in the previous paragraph.

There is also the danger of invalidating the input format while removing features. In this thesis, the input consists of the surface forms of a sentence along with the possible morphological analyses of the words it embodies. When we remove a morphological feature from an analysis, we risk modifying the analysis so that it is no longer conforming to the rules of expressing a morphological analysis in the relevant scheme. For example, the removal of morphological feature 'P3sg' ('the noun is possessed by the third singular') invalidates the morphological analysis of a noun because every noun should have a possessive marker. This could cause the model to misinterpret the morphological analysis as it might expect the analysis to conform to a certain syntax. Instead, it would be better to determine an appropriate approach for meaningfully removing a single or a group of features. For example, the correct way of removing a possessive marker such as 'P3sg' is to replace it with 'Pnon' which indicates no possession. However, this is not an easy task for our case as a modification in the morphological analysis is usually reflected in the surface form, e.g. we should also remove a '-sı' suffix from the word 'arabası' ('his/her car') if we remove the morphological feature 'P3sg'. This difficulty stems from the textual representation of natural language. In comparison to visual input, textual input has no physical analogy. Visual medium is composed of lines, corners, simple shapes, color gradients, shadows, and others. These are all observed in the nature with the naked eye, and thus gov-

erned with the laws of optical physics. On the other hand, written text is composed of symbols. The shapes that make up these symbols typically do not bear similarity to the things they represent, i.e. the graphical representation of the word 'person' is unrelated to the figure of 'person' in the minds of people. There are some exceptions to this observation in some writing systems, i.e. the Chinese character meaning 'person' or 'man' looks like a walking man. But in general written text is an arbitrary encoding of meaning. So they are very dependent on each other, i.e. they must follow certain rules of the language. When we change a letter in an English word or add a strike to a Chinese character, the meaning may change in unexpected ways. In contrast, visual input is more robust in the sense that changing specific features (pixels) in the input does not invalidate the input. Invalidation is only a danger when large portions are modified to contain random noise or an object which does not conform to the optical rules imposed by the camera and the prejudices of human vision. As a result, the family of explanation methods that merely remove a feature from or nullify a feature in the input to explain the outcome of a model aligns better with visual input. Further work on automatically determining which parts of a textual input should be modified would be very interesting and valuable.

The evaluation of explanation methods is challenging. The main problem is rooted in the difficulty of defining what makes an explanation high-quality. The difficulty arises because every explanation is built based on a reduced model of the inner workings of the model. As the model itself is based on an approximate view of the problem that it aims to solve, the connection between the explanation and the problem is weak. Suppose that an experienced engineer needs to model a bridge to determine whether it would collapse under certain loads. The input to the model is the current number of automobiles on the bridge, load readings at several locations on the bridge, and a detailed blue-print of the bridge in machine-readable format. The explanation of a positive output of this model could be a heat-map on the blueprint indicating the parts that are under high-load. This explanation may be a suitable high-quality explanation since it can be tied to an analytical view of the bridge that can be evaluated using the actual measurements. Nonetheless, only people with domain expertise can read it and assess its plausibility. Furthermore, although the explanation seems

plausible to an engineer, it might be the case that the model in question is paying more attention to the number of automobiles on the bridge than the information that the blue-print of the bridge conveys. If this is the case, the explanation method is not faithful because it portrays the model as if it is employing an approach that people with domain-knowledge would use while it is instead just focusing on the number of automobiles on the bridge. As we examined this example explanation method, we have touched upon three aspects of an explanation of high-quality: *readability*, *plausibility*, and *faithfulness* [127].

The evaluation of the explanation method we propose is examined according the above described criteria, namely readability, plausibility, and faithfulness. Recall that an explanation in our case is simply a list of features and their corresponding importance values. The readability is high since it explicitly presents the language features. To evaluate the plausibility of explanations, we calculated the average of the importance values of each feature for every named entity tag. The rankings of the average scores for each morphological feature with respect to its importance for in predicting each entity tag types are shown in Table 5.4 and 5.7 for Finnish and Turkish respectively. When we observe the top and bottom features in these tables, we see that the explanations lead us to plausible explanations in some of the named entities, e.g. features related to location semantics are the top ones for location named entity. Additionally, the analysis that was done by processing the rules of FiNER NER tagger supports the plausibility claim as the top morphological features presented in Table 5.4 matches the ones that are found to lead to named entities in FiNER rules. Our evaluation does not address the faithfulness aspect. However, we believe that our explanation method (EXSEQREG) can be considered faithful. We expect a model to employ every feature in the input. Furthermore, EXSEQREG focuses on the changes that are induced by removal of only a subset of all features (morphological feature tags). The features that do not induce a change in the probability are easily recognizable as they would have an importance value of zero. If, instead of the current approach, EXSEQREG was a method that learns to output its own explanations using the input and output of the model, unfaithfulness would be a risk.

# 7. CONCLUSION

In this thesis, we introduce a NER tagger that incorporates morphological information to improve the performance for morphologically rich languages. Our experiments show that disambiguated morphological analyses can be utilized for this purpose to surpass state-of-the-art performance levels for Turkish, Czech, Hungarian, Finnish, and Spanish. It is shown that augmenting word representations with morphological and character embeddings is responsible for this improvement. We also show that it is possible to perform morphological disambiguation jointly in a single neural network. This eliminates the dependency for an external morphological disambiguator, making it easier to port the NER tagger to other languages.

The second thread of this thesis proposes a model-agnostic explanation method for explaining sequence-based NLP tasks. This method is an adaptation of an explanation method to the NLP context. This method formalizes the concept of a region within an input sample and its relation to an output label. Our method lends itself to extensions to accommodate additional NLP tasks with differing input and output formats. Finally, as this formalization can be used for further research, it has the potential to facilitate further research on explanation methods for NLP.

We publish the NER tagger as open source software which has received positive feedback from the software development industry. Numerous correspondences have occurred with respect to integrating them into NLP systems. In addition to the open source platforms that make this work available, it will also be served on an NLP platform being developed at Bogazici University.

# REFERENCES


1. Grishman, R. and B. M. Sundheim, "Message understanding conference-6: A brief history", *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*, 1996.

2. Kripke, S. A., "Naming and necessity", *Semantics of Natural Language*, pp. 253–355, Springer, 1972.

3. Lample, G., M. Ballesteros, S. Subramanian, K. Kawakami and C. Dyer, "Neural Architectures for Named Entity Recognition", *Proceedings of NAACL-HLT*, pp. 260–270, 2016.

4. Huang, Z., W. Xu and K. Yu, "Bidirectional LSTM-CRF models for sequence tagging", *arXiv preprint arXiv:1508.01991*, 2015.

5. Ma, X. and E. Hovy, "End-to-end Sequence Labeling via Bi-directional LSTM-CNNs-CRF", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1064–1074, Association for Computational Linguistics, Berlin, Germany, Aug. 2016.

6. Yang, Z., R. Salakhutdinov and W. Cohen, "Multi-task cross-lingual sequence tagging from scratch", *arXiv preprint arXiv:1603.06270*, 2016.

7. Oflazer, K., "Two-level description of Turkish morphology", *Literary and Linguistic Computing*, Vol. 9, No. 2, pp. 137–148, 1994.

8. Kuru, O., O. A. Can and D. Yuret, "CharNER: Character-level named entity recognition", *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 911–921, 2016.

9. Tür, G., D. Hakkani-Tür and K. Oflazer, "A statistical information extraction

system for Turkish", *Natural Language Engineering*, Vol. 9, No. 2, pp. 181–210, 2003.

10. Yeniterzi, R., "Exploiting Morphology in Turkish Named Entity Recognition System", *Proceedings of the ACL 2011 Student Session*, HLT-SS '11, pp. 105–110, Association for Computational Linguistics, Stroudsburg, PA, USA, 2011.

11. Güngör, O., S. Üsküdarlı and T. Güngör, "Recurrent neural networks for Turkish named entity recognition", *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2018.

12. Güngör, O., T. Güngör and S. Üsküdarlı, "The effect of morphology in named entity recognition with sequence tagging", *Natural Language Engineering*, Vol. 25, No. 1, p. 147–169, 2019.

13. Güngör, O., S. Üsküdarlı and T. Güngör, "Improving Named Entity Recognition by Jointly Learning to Disambiguate Morphological Tags", *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2082–2092, Association for Computational Linguistics, Santa Fe, New Mexico, USA, Aug. 2018.

14. Güngör, O., T. Güngör and S. Uskudarli, "EXSEQREG: Explaining sequence-based NLP tasks with regions with a case study using morphological features for named entity recognition", *PLOS ONE*, Vol. 15, No. 12, pp. 1–29, 12 2020.

15. Miwa, M. and M. Bansal, "End-to-End Relation Extraction using LSTMs on Sequences and Tree Structures", *CoRR*, Vol. abs/1601.00770, 2016.

16. Rao, D., P. McNamee and M. Dredze, "Entity linking: Finding extracted entities in a knowledge base", *Multi-source, multilingual information extraction and summarization*, pp. 93–115, Springer, 2013.

17. Lee, J., G. Kim, J. Yoo, C. Jung, M. Kim and S. Yoon, "Training IBM

Watson Using Automatically Generated Question-Answer Pairs", *CoRR*, Vol. abs/1611.03932, 2017.

18. Liu, Y. and F. Ren, "Japanese named entity recognition for question answering system", *2011 IEEE International Conference on Cloud Computing and Intelligence Systems*, pp. 402–406, IEEE, 2011.

19. Lee, C., Y.-G. Hwang and M.-G. Jang, "Fine-grained named entity recognition and relation extraction for question answering", *Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 799–800, 2007.

20. Humphreys, K., R. Gaizauskas, S. Azzam, C. Huyck, B. Mitchell, H. Cunningham and Y. Wilks, "University of Sheffield: Description of the LaSIE-II system as used for MUC-7", *Proceedings of the Seventh Message Understanding Conferences (MUC-7)*, ACL, 1998.

21. Appelt, D. E., J. R. Hobbs, J. Bear, D. Israel, M. Kameyama, D. Martin, K. Myers and M. Tyson, "SRI International FASTUS system: MUC-6 test results and analysis", *Proceedings of the 6th Conference on Message Understanding*, pp. 237–248, Association for Computational Linguistics, 1995.

22. McCallum, A. and W. Li, "Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons", *Proceedings of the 7th Conference on Natural Language Learning at HLT-NAACL*, Vol. 4, pp. 188–191, Association for Computational Linguistics, 2003.

23. Finkel, J. R., T. Grenager and C. Manning, "Incorporating non-local information into information extraction systems by Gibbs sampling", *Proceedings of the 43rd Annual Meeting of ACL*, pp. 363–370, Association for Computational Linguistics, 2005.

24. Borthwick, A. E., *A Maximum Entropy Approach to Named Entity Recognition*,

Ph.D. Thesis, New York University, New York, NY, USA, 1999.

25. Jiang, J. and C. X. Zhai, "Instance weighting for domain adaptation in NLP", *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Vol. 7, pp. 264–271, ACL, Prague, 2007.

26. Wu, D., W. S. Lee, N. Ye and H. L. Chieu, "Domain adaptive bootstrapping for named entity recognition", *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 3-Volume 3*, pp. 1523–1532, Association for Computational Linguistics, 2009.

27. Guo, H., H. Zhu, Z. Guo, X. Zhang, X. Wu and Z. Su, "Domain adaptation with latent semantic association for named entity recognition", *Proceedings of Human Language Technologies: The 2009 Annual Conference of the NAACL*, pp. 281–289, ACL, 2009.

28. Szarvas, G., R. Farkas and A. Kocsor, "A multilingual named entity recognition system using boosting and C4.5 decision tree learning algorithms", *International Conference on Discovery Science*, pp. 267–278, Springer, 2006.

29. Collobert, R., J. Weston, L. Bottou, M. Karlen, K. Kavukcuoglu and P. Kuksa, "Natural language processing (almost) from scratch", *Journal of Machine Learning Research*, Vol. 12, No. Aug, pp. 2493–2537, 2011.

30. Harris, Z. S., "Distributional structure", *Word*, Vol. 10, No. 2-3, pp. 146–162, 1954.

31. Socher, R., A. Perelygin, J. Wu, J. Chuang, C. D. Manning, A. Y. Ng and C. Potts, "Recursive deep models for semantic compositionality over a sentiment treebank", *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1631–1642, 2013.

32. Collobert, R. and J. Weston, "A unified architecture for natural language pro-

cessing: Deep neural networks with multitask learning", *Proceedings of the 25th International Conference on Machine Learning*, pp. 160–167, 2008.

33. Mikolov, T., M. Karafiát, L. Burget, J. Černockỳ and S. Khudanpur, "Recurrent neural network based language model", *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

34. Mikolov, T., I. Sutskever, K. Chen, G. S. Corrado and J. Dean, "Distributed representations of words and phrases and their compositionality", *Advances in Neural Information Processing Systems*, pp. 3111–3119, 2013.

35. Pennington, J., R. Socher and C. D. Manning, "GloVe: Global vectors for word representation", *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 1532–1543, 2014.

36. Luong, M.-T., R. Socher and C. D. Manning, "Better word representations with recursive neural networks for morphology", *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pp. 104–113, 2013.

37. Santos, C. D. and B. Zadrozny, "Learning character-level representations for part-of-speech tagging", *Proceedings of the 31st International Conference on Machine Learning (ICML-14)*, pp. 1818–1826, 2014.

38. Xu, Y. and J. Liu, "Implicitly incorporating morphological information into word embedding", *arXiv preprint arXiv:1701.02481*, 2017.

39. Bhatia, P., R. Guthrie and J. Eisenstein, "Morphological priors for probabilistic neural word embeddings", *arXiv preprint arXiv:1608.01056*, 2016.

40. Lankinen, M., H. Heikinheimo, P. Takala, T. Raiko and J. Karhunen, "A character-word compositional neural language model for Finnish", *arXiv preprint arXiv:1612.03266*, 2016.

41. Shen, Q., D. Clothiaux, E. Tagtow, P. Littell and C. Dyer, "The role of context in neural morphological disambiguation", *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pp. 181–191, 2016.

42. Cotterell, R. and H. Schütze, "Joint semantic synthesis and morphological analysis of the derived word", *Transactions of the Association for Computational Linguistics*, Vol. 6, pp. 33–48, 2018.

43. Demir, H. and A. Özgür, "Improving named entity recognition for morphologically rich languages using word embeddings", *13th International Conference on Machine Learning and Applications*, pp. 117–122, IEEE, 2014.

44. Seker, G. A. and G. Eryiğit, "Initial Explorations on using CRFs for Turkish Named Entity Recognition", *Proceedings of COLING 2012*, pp. 2459–2474, The COLING 2012 Organizing Committee, Mumbai, India, December 2012.

45. Hasan, K. S., M. A. u. Rahman and V. Ng, "Learning-Based Named Entity Recognition for Morphologically-Rich, Resource-Scarce Languages", *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pp. 354–362, Association for Computational Linguistics, Athens, Greece, Mar. 2009.

46. Freund, Y. and R. E. Schapire, "Large margin classification using the perceptron algorithm", *Machine Learning*, Vol. 37, No. 3, pp. 277–296, 1999.

47. Yildiz, E., C. Tirkaz, H. B. Sahin, M. T. Eren and O. O. Sonmez, "A Morphology-Aware Network for Morphological Disambiguation", *30th AAAI Conference on Artificial Intelligence*, 2016.

48. Silfverberg, M., T. Ruokolainen, K. Lindén and M. Kurimo, "FinnPos: an open-source morphological tagging and lemmatization toolkit for Finnish", *Language Resources and Evaluation*, Vol. 50, No. 4, pp. 863–878, 2016.

49. Trón, V., A. Kornai, G. Gyepesi, L. Németh, P. Halácsy and D. Varga, "Hunmorph: open source word analysis", *Proceedings of the Workshop on Software*, pp. 77–85, Association for Computational Linguistics, 2005.

50. Nivre, J., M.-C. de Marneffe, F. Ginter, Y. Goldberg, J. Hajic, C. D. Manning, R. T. McDonald, S. Petrov, S. Pyysalo, N. Silveira *et al.*, "Universal Dependencies v1: A Multilingual Treebank Collection.", *International Conference on Language Resources and Evaluation (LREC)*, 2016.

51. Hashimoto, K., C. Xiong, Y. Tsuruoka and R. Socher, "A Joint Many-Task Model: Growing a Neural Network for Multiple NLP Tasks", *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2017.

52. Luong, M.-T., Q. V. Le, I. Sutskever, O. Vinyals and L. Kaiser, "Multi-task sequence to sequence learning", *arXiv preprint arXiv:1511.06114*, 2015.

53. Søgaard, A. and Y. Goldberg, "Deep multi-task learning with low level tasks supervised at lower layers", *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pp. 231–235, Association for Computational Linguistics, Berlin, Germany, Aug. 2016.

54. Guidotti, R., A. Monreale, S. Ruggieri, F. Turini, F. Giannotti and D. Pedreschi, "A survey of methods for explaining black box models", *ACM Computing Surveys (CSUR)*, Vol. 51, No. 5, p. 93, 2018.

55. Ribeiro, M. T., S. Singh and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier", *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 1135–1144, ACM, 2016.

56. Guidotti, R., A. Monreale, S. Ruggieri, D. Pedreschi, F. Turini and F. Giannotti, "Local rule-based explanations of black box decision systems", *arXiv preprint arXiv:1805.10820*, 2018.

57. Lundberg, S. M. and S.-I. Lee, "A unified approach to interpreting model predictions", *Advances in Neural Information Processing Systems*, pp. 4765–4774, 2017.

58. Shrikumar, A., P. Greenside and A. Kundaje, "Learning important features through propagating activation differences", *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3145–3153, Feb. 2017.

59. Bach, S., A. Binder, G. Montavon, F. Klauschen, K.-R. Müller and W. Samek, "On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation", *PLOS ONE*, Vol. 10, No. 7, p. e0130140, 2015.

60. Datta, A., S. Sen and Y. Zick, "Algorithmic transparency via quantitative input influence: Theory and experiments with learning systems", *IEEE Symposium on Security and Privacy (SP)*, pp. 598–617, IEEE, 2016.

61. Lipovetsky, S. and M. Conklin, "Analysis of regression in game theory approach", *Applied Stochastic Models in Business and Industry*, Vol. 17, No. 4, pp. 319–330, 2001.

62. Štrumbelj, E. and I. Kononenko, "Explaining prediction models and individual predictions with feature contributions", *Knowledge and Information Systems*, Vol. 41, No. 3, pp. 647–665, 2014.

63. Villarroya, S. H., *Interpretability in sequence tagging models for Named Entity Recognition*, Master's Thesis, University of Amsterdam, 2018.

64. Poerner, N., H. Schütze and B. Roth, "Evaluating neural network explanation methods using hybrid documents and morphological prediction", *56th Annual Meeting of the Association for Computational Linguistics (ACL)*, 2018.

65. Adi, Y., E. Kermany, Y. Belinkov, O. Lavi and Y. Goldberg, "Fine-grained analysis of sentence embeddings using auxiliary prediction tasks", *arXiv preprint*

*arXiv:1608.04207*, 2016.

66. Alishahi, A., M. Barking and G. Chrupała, "Encoding of phonology in a recurrent neural model of grounded speech", *Proceedings of the Conference on Natural Language Learning (CoNLL)*, p. 368, 2017.

67. Conneau, A., G. Kruszewski, G. Lample, L. Barrault and M. Baroni, "What you can cram into a single\ &!#* vector: Probing sentence embeddings for linguistic properties", *56th Annual Meeting of the Association for Computational Linguistics*, pp. 2126–2136, 2018.

68. Hupkes, D., S. Veldhoen and W. Zuidema, "Visualisation and diagnostic classifiers reveal how recurrent and recursive neural networks process hierarchical structure", *Journal of Artificial Intelligence Research*, Vol. 61, pp. 907–926, 2018.

69. Li, J., W. Monroe and D. Jurafsky, "Understanding neural networks through representation erasure", *arXiv preprint arXiv:1612.08220*, 2016.

70. Jacovi, A., O. S. Shalom and Y. Goldberg, "Understanding Convolutional Neural Networks for Text Classification", *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 56–65, 2018.

71. Poerner, N., B. Roth and H. Schütze, "Interpretable Textual Neuron Representations for NLP", *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 325–327, 2018.

72. Burns, K., A. Nematzadeh, E. Grant, A. Gopnik and T. Griffiths, "Exploiting attention to reveal shortcomings in memory models", *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 378–380, 2018.

73. Sommerauer, P. and A. Fokkens, "Firearms and tigers are dangerous, kitchen knives and zebras are not: testing whether word embeddings can tell", *Proceed-*

*ings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 276–286, 2018.

74. Yin, P., C. Zhou, J. He and G. Neubig, "StructVAE: Tree-structured Latent Variable Models for Semi-supervised Semantic Parsing", *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 754–765, 2018.

75. Stahlberg, F., D. Saunders and B. Byrne, "An Operation Sequence Model for Explainable Neural Machine Translation", *Proceedings of the EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pp. 175–186, 2018.

76. Ramshaw, L. A. and M. P. Marcus, "Text Chunking using Transformation-Based Learning", *CoRR*, Vol. cmp-lg/9505040, 1995.

77. Koskenniemi, K., *Two-level morphology: A general computational model for word form recognition and production*, Ph.D. Thesis, Department of General Linguistics, University of Helsinki, 1983.

78. Underhill, R., *Turkish Grammar*, MIT press Cambridge, MA, 1976.

79. Lewis, G. L., *Turkish Grammar*, Oxford University Press, Oxford, 1991.

80. Oflazer, K., "Dependency parsing with an extended finite-state approach", *Computational Linguistics*, Vol. 29, No. 4, pp. 515–544, 2003.

81. Silfverberg, M., T. Ruokolainen, K. Lindén and M. Kurimo, "FinnPos: an open-source morphological tagging and lemmatization toolkit for Finnish", *Language Resources and Evaluation*, Vol. 50, No. 4, pp. 863–878, 2016.

82. Pirinen, T. A., "Omorfi—Free and open source morphological lexical database for Finnish", *Proceedings of the 20th Nordic Conference of Computational Linguistics*

*(NODALIDA 2015)*, pp. 313–315, 2015.

83. Ševčíková, M., Z. Žabokrtský and O. Krůza, "Named entities in Czech: annotating data and developing NE tagger", *International Conference on Text, Speech and Dialogue*, pp. 188–195, Springer, 2007.

84. Hajič, J., J. Panevová, E. Hajičová, P. Sgall, P. Pajas, J. Štěpánek, J. Havelka, M. Mikulová, Z. Žabokrtský, M. Ševčíková Razímová and Z. Urešová, *Prague Dependency Treebank 2.0*, Linguistic Data Consortium, Philadelphia, PA, USA, 2006.

85. Hajič, J., E. Hajičová, M. Mikulová and J. Mírovský, "Prague dependency treebank", *Handbook of Linguistic Annotation*, pp. 555–594, Springer, 2017.

86. Hana, J., D. Zeman, J. Hajic, H. Hanová, B. Hladká and E. Jerábek, "Manual for Morphological Annotation. Revision for the Prague Dependency Treebank 2.0", *UFAL MFF UK, Prague*, 2005.

87. Zsibrita, J., V. Vincze and R. Farkas, "magyarlanc: A tool for morphological and dependency parsing of Hungarian", *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pp. 763–771, 2013.

88. Prószéky, G. and L. Tihanyi, "Humor: High-speed unification morphology and its applications for agglutinative languages", *La Tribune des Industries de la Langue*, Vol. 10, pp. 28–29, 1993.

89. Erjavec, T., "MULTEXT-East Version 4: Multilingual Morphosyntactic Specifications, Lexicons and Corpora.", *International Conference on Language Resources and Evaluation (LREC)*, 2010.

90. Rebrus, P., A. Kornai and P. Vajda, "The annotation system of HunMorph", http://real.mtak.hu/id/eprint/24283, 2006.

91. Farkas, R., D. Szeredi, D. Varga and V. Vincze, "MSD-KR harmonizáció a Szeged Treebank 2.5-ben [Harmonizing MSD and KR codes in the Szeged Treebank 2.5]", *VII. Magyar Számıtógépes Nyelvészeti Konferencia*, pp. 349–353, 2010.

92. Hochreiter, S. and J. Schmidhuber, "Long short-term memory", *Neural Computation*, Vol. 9, No. 8, pp. 1735–1780, 1997.

93. Graves, A. and J. Schmidhuber, "Framewise phoneme classification with bidirectional LSTM networks", *Proceedings of IEEE International Joint Conference on Neural Networks*, Vol. 4, pp. 2047–2052, IEEE, 2005.

94. Mozer, M. C., "A Focused Backpropagation Algorithm for Temporal Pattern Recognition", *Complex Systems*, Vol. 3, No. 4, pp. 349–381, 1989.

95. Kingma, D. P. and J. Ba, "Adam: A Method for Stochastic Optimization", Y. Bengio and Y. LeCun (Editors), *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015.

96. Tieleman, T. and G. Hinton, "Lecture 6.5—RmsProp: Divide the gradient by a running average of its recent magnitude", COURSERA: Neural Networks for Machine Learning, 2012.

97. Lafferty, J. D., A. McCallum and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, p. 282–289, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2001.

98. Turian, J., L. Ratinov and Y. Bengio, "Word representations: a simple and general method for semi-supervised learning", *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pp. 384–394, Association for Computational Linguistics, 2010.

99. Bojanowski, P., E. Grave, A. Joulin and T. Mikolov, "Enriching word vectors with subword information", *Transactions of the Association for Computational Linguistics*, Vol. 5, pp. 135–146, 2017.

100. Votrubec, J., "Morphological tagging based on averaged perceptron", *WDS'06 Proceedings of Contributed Papers*, pp. 191–195, 2006.

101. Güngör, O. and E. Yıldız, "Linguistic features in Turkish word representations", *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, IEEE, 2017.

102. Konkol, M. and M. Konopík, "CRF-based Czech named entity recognizer and consolidation of Czech NER research", *International Conference on Text, Speech and Dialogue*, pp. 153–160, Springer, 2013.

103. Szarvas, G., R. Farkas, L. Felföldi, A. Kocsor and J. Csirik, "A highly accurate Named Entity corpus for Hungarian.", *International Conference on Language Resources and Evaluation (LREC)*, pp. 1957–1960, 2006.

104. Ruokolainen, T., P. Kauppinen, M. Silfverberg and K. Lindén, "A Finnish news corpus for named entity recognition", *Language Resources and Evaluation*, Vol. 54, No. 1, pp. 247–272, 2020.

105. Schweter, S., "BERTurk - BERT models for Turkish", https://doi.org/10.5281/zenodo.3770924, Apr. 2020, https://doi.org/10.5281/zenodo.3770924.

106. Varga, D. and E. Simon, "Hungarian named entity recognition with a maximum entropy approach", *Acta Cybernetica*, Vol. 18, No. 2, pp. 293–301, 2007.

107. Straková, J., M. Straka and J. Hajič, "Neural networks for featureless named entity recognition in Czech", *International Conference on Text, Speech, and Dialogue*, pp. 173–181, Springer, 2016.

108. Straka, M. and J. Straková, "Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe", *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pp. 88–99, 2017.

109. Nair, V. and G. E. Hinton, "Rectified linear units improve restricted Boltzmann machines", *Proceedings of the 27th international conference on machine learning (ICML-10)*, pp. 807–814, 2010.

110. Neubig, G., C. Dyer, Y. Goldberg, A. Matthews, W. Ammar, A. Anastasopoulos, M. Ballesteros, D. Chiang, D. Clothiaux, T. Cohn, K. Duh, M. Faruqui, C. Gan, D. Garrette, Y. Ji, L. Kong, A. Kuncoro, G. Kumar, C. Malaviya, P. Michel, Y. Oda, M. Richardson, N. Saphra, S. Swayamdipta and P. Yin, "DyNet: The Dynamic Neural Network Toolkit", *arXiv preprint arXiv:1701.03980*, 2017.

111. Kingma, D. and J. Ba, "Adam: A method for stochastic optimization", *arXiv preprint arXiv:1412.6980*, 2014.

112. Srivastava, N., G. E. Hinton, A. Krizhevsky, I. Sutskever and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting.", *Journal of Machine Learning Research*, Vol. 15, No. 1, pp. 1929–1958, 2014.

113. Reimers, N. and I. Gurevych, "Optimal Hyperparameters for Deep LSTM-Networks for Sequence Labeling Tasks", *arXiv preprint arXiv:1707.06799*, 2017.

114. Yuret, D. and F. Türe, "Learning morphological disambiguation rules for Turkish", *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pp. 328–334, Association for Computational Linguistics, 2006.

115. Shrikumar, A., P. Greenside, A. Shcherbina and A. Kundaje, "Not just a black box: Learning important features through propagating activation differences", *arXiv preprint arXiv:1605.01713*, 2016.

116. Selvaraju, R. R., M. Cogswell, A. Das, R. Vedantam, D. Parikh and D. Batra, "Grad-cam: Visual explanations from deep networks via gradient-based localization", *Proceedings of the IEEE International Conference on Computer Vision*, pp. 618–626, 2017.

117. Güngör, O., "Code for conducting the experiments", https://github.com/onurgu/joint-ner-and-md-tagger/tree/xnlp, 2019.

118. University of Helsinki, "Finnish News Corpus for Named Entity Recognition", http://urn.fi/urn:nbn:fi:lb-2019050201, 2019.

119. Nivre, J., M. Abrams, Ž. Agić, L. Ahrenberg, G. Aleksandravičiūtė, L. Antonsen, K. Aplonova, M. J. Aranzabe, G. Arutie, M. Asahara, L. Ateyah, M. Attia, A. Atutxa, L. Augustinus, E. Badmaeva, M. Ballesteros, E. Banerjee, S. Bank, V. Barbu Mititelu, V. Basmov, J. Bauer, S. Bellato, K. Bengoetxea, Y. Berzak, I. A. Bhat, R. A. Bhat, E. Biagetti, E. Bick, A. Bielinskienė, R. Blokland, V. Bobicev, L. Boizou, E. Borges Völker, C. Börstell, C. Bosco, G. Bouma, S. Bowman, A. Boyd, K. Brokaitė, A. Burchardt, M. Candito, B. Caron, G. Caron, G. Cebiroğlu Eryiğit, F. M. Cecchini, G. G. A. Celano, S. Čéplö, S. Cetin, F. Chalub, J. Choi, Y. Cho, J. Chun, S. Cinková, A. Collomb, Ç. Çöltekin, M. Connor, M. Courtin, E. Davidson, M.-C. de Marneffe, V. de Paiva, A. Diaz de Ilarraza, C. Dickerson, B. Dione, P. Dirix, K. Dobrovoljc, T. Dozat, K. Droganova, P. Dwivedi, H. Eckhoff, M. Eli, A. Elkahky, B. Ephrem, T. Erjavec, A. Etienne, R. Farkas, H. Fernandez Alcalde, J. Foster, C. Freitas, K. Fujita, K. Gajdošová, D. Galbraith, M. Garcia, M. Gärdenfors, S. Garza, K. Gerdes, F. Ginter, I. Goenaga, K. Gojenola, M. Gökırmak, Y. Goldberg, X. Gómez Guinovart, B. González Saavedra, M. Grioni, N. Grūzītis, B. Guillaume, C. Guillot-Barbance, N. Habash, J. Hajič, J. Hajič jr., L. Hà Mỹ, N.-R. Han, K. Harris, D. Haug, J. Heinecke, F. Hennig, B. Hladká, J. Hlaváčová, F. Hociung, P. Hohle, J. Hwang, T. Ikeda, R. Ion, E. Irimia, Ọ. Ishola, T. Jelínek, A. Johannsen, F. Jørgensen, H. Kaşıkara, A. Kaasen, S. Kahane, H. Kanayama, J. Kanerva, B. Katz, T. Kayadelen, J. Kenney, V. Kettnerová,

J. Kirchner, A. Köhn, K. Kopacewicz, N. Kotsyba, J. Kovalevskaitė, S. Krek, S. Kwak, V. Laippala, L. Lambertino, L. Lam, T. Lando, S. D. Larasati, A. Lavrentiev, J. Lee, P. Lê H`ông, A. Lenci, S. Lertpradit, H. Leung, C. Y. Li, J. Li, K. Li, K. Lim, Y. Li, N. Ljubešić, O. Loginova, O. Lyashevskaya, T. Lynn, V. Macketanz, A. Makazhanov, M. Mandl, C. Manning, R. Manurung, C. Mărănduc, D. Mareček, K. Marheinecke, H. Martínez Alonso, A. Martins, J. Mašek, Y. Matsumoto, R. McDonald, S. McGuinness, G. Mendonça, N. Miekka, M. Misirpashayeva, A. Missilä, C. Mititelu, Y. Miyao, S. Montemagni, A. More, L. Moreno Romero, K. S. Mori, T. Morioka, S. Mori, S. Moro, B. Mortensen, B. Moskalevskyi, K. Muischnek, Y. Murawaki, K. Müürisep, P. Nainwani, J. I. Navarro Horñiacek, A. Nedoluzhko, G. Nešpore-Bērzkalne, L. Nguy˜ên Thị, H. Nguy˜ên Thị Minh, Y. Nikaido, V. Nikolaev, R. Nitisaroj, H. Nurmi, S. Ojala, A. Olúòkun, M. Omura, P. Osenova, R. Östling, L. Øvrelid, N. Partanen, E. Pascual, M. Passarotti, A. Patejuk, G. Paulino-Passos, A. Peljak-Łapińska, S. Peng, C.-A. Perez, G. Perrier, D. Petrova, S. Petrov, J. Piitulainen, T. A. Pirinen, E. Pitler, B. Plank, T. Poibeau, M. Popel, L. Pretkalniņa, S. Prévost, P. Prokopidis, A. Przepiórkowski, T. Puolakainen, S. Pyysalo, A. Rääbis, A. Rademaker, L. Ramasamy, T. Rama, C. Ramisch, V. Ravishankar, L. Real, S. Reddy, G. Rehm, M. Rießler, E. Rimkutė, L. Rinaldi, L. Rituma, L. Rocha, M. Romanenko, R. Rosa, D. Rovati, V. Roșca, O. Rudina, J. Rueter, S. Sadde, B. Sagot, S. Saleh, A. Salomoni, T. Samardžić, S. Samson, M. Sanguinetti, D. Särg, B. Saulīte, Y. Sawanakunanon, N. Schneider, S. Schuster, D. Seddah, W. Seeker, M. Seraji, M. Shen, A. Shimada, H. Shirasu, M. Shohibussirri, D. Sichinava, N. Silveira, M. Simi, R. Simionescu, K. Simkó, M. Šimková, K. Simov, A. Smith, I. Soares-Bastos, C. Spadine, A. Stella, M. Straka, J. Strnadová, A. Suhr, U. Sulubacak, S. Suzuki, Z. Szántó, D. Taji, Y. Takahashi, F. Tamburini, T. Tanaka, I. Tellier, G. Thomas, L. Torga, T. Trosterud, A. Trukhina, R. Tsarfaty, F. Tyers, S. Uematsu, Z. Urešová, L. Uria, H. Uszkoreit, S. Vajjala, D. van Niekerk, G. van Noord, V. Varga, E. Villemonte de la Clergerie, V. Vincze, L. Wallin, A. Walsh, J. X. Wang, J. N. Washington, M. Wendt, S. Williams, M. Wirén, C. Wittern, T. Woldemariam, T.-s. Wong, A. Wróblewska, M. Yako,

N. Yamazaki, C. Yan, K. Yasuoka, M. M. Yavrumyan, Z. Yu, Z. Žabokrtský, A. Zeldes, D. Zeman, M. Zhang and H. Zhu, "Universal Dependencies 2.4", , 2019, http://hdl.handle.net/11234/1-2988, LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

120. Güngör, O., "Branch of UDPipe: Trainable pipeline for tokenizing, tagging, lemmatizing and parsing Universal Treebanks and other CoNLL-U files", https://github.com/onurgu/udpipe, 2018.

121. Straka, M., J. Hajic and J. Straková, "UDPipe: Trainable Pipeline for Processing CoNLL-U Files Performing Tokenization, Morphological Analysis, POS Tagging and Parsing.", *Proceedings of the 10th International Conference on Language Resources and Evaluation (LREC 2016)*, pp. 4290–4297, 2016.

122. Alishahi, A., G. Chrupala and T. Linzen, "Analyzing and Interpreting Neural Networks for NLP: A Report on the First BlackboxNLP Workshop", *Natural Language Engineering*, 2019.

123. University of Helsinki, "Finnish Tag Tools", http://urn.fi/urn:nbn:fi:lb-201811141, 2019.

124. Karttunen, L., "Pattern Matching with FST — A Tutorial", https://web.stanford.edu/~laurik/publications/pmatch.pdf, 2010.

125. Devlin, J., M.-W. Chang, K. Lee and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding", *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4171–4186, Association for Computational Linguistics, Minneapolis, Minnesota, Jun. 2019.

126. Peters, M., M. Neumann, M. Iyyer, M. Gardner, C. Clark, K. Lee and L. Zettle-

moyer, "Deep Contextualized Word Representations", *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pp. 2227–2237, Association for Computational Linguistics, New Orleans, Louisiana, Jun. 2018.

127. Jacovi, A. and Y. Goldberg, "Towards Faithfully Interpretable NLP Systems: How Should We Define and Evaluate Faithfulness?", *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4198–4205, Association for Computational Linguistics, Online, Jul. 2020.

# APPENDIX A: ADDITIONAL TABLES REGARDING CHAPTER 5

Table A.1. The ranks of $\hat{\mu}_k(m)$ for each entity tag in Finnish. The morphological tags that are in the first 10 for at least one entity tag are shown. (1/3)

| Morphological Tag | ORG | TIT | PER | TIM | LOC | DATE | PRO | MISC | EVENT | OUTSIDE | median | std. dev. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Number=Sing | 89 | 1 | 89 | 89 | 89 | 88 | 89 | 2 | 3 | 89 | 89 | 42 |
| Case=Nom | 88 | 89 | 3 | 82 | 88 | 11 | 1 | 89 | 1 | 1 | 3 | 39 |
| Voice=Act | 3 | 83 | 5 | 80 | 79 | 6 | 12 | 83 | 84 | 85 | 12 | 30 |
| VerbForm=Fin | 6 | 79 | 9 | 61 | 8 | 12 | 8 | 78 | 12 | 86 | 8 | 26 |
| *BLANK* | 63 | 42 | 41 | 40 | 44 | 78 | 63 | 27 | 37 | 29 | 44 | 18 |
| Mood=Ind | 8 | 81 | 70 | 52 | 9 | 14 | 9 | 81 | 16 | 88 | 9 | 27 |
| Number=Plur | 4 | 85 | 4 | 86 | 3 | 23 | 78 | 1 | 89 | 75 | 4 | 35 |
| Case=Gen | 87 | 87 | 88 | 79 | 2 | 7 | 88 | 13 | 87 | 2 | 87 | 37 |
| Degree=Pos | 2 | 88 | 85 | 2 | 87 | 1 | 82 | 84 | 6 | 3 | 82 | 41 |
| Person=3 | 20 | 84 | 7 | 54 | 17 | 33 | 76 | 66 | 28 | 83 | 20 | 19 |
| Tense=Pres | 10 | 9 | 16 | 68 | 10 | 13 | 14 | 76 | 14 | 87 | 14 | 32 |
| Case=Par | 17 | 86 | 87 | 85 | 6 | 86 | 2 | 85 | 7 | 14 | 7 | 28 |
| Tense=Past | 12 | 80 | 76 | 21 | 15 | 45 | 10 | 72 | 78 | 12 | 15 | 26 |
| Style=Coll | 14 | 74 | 2 | 9 | 21 | 32 | 16 | 50 | 32 | 22 | 16 | 22 |
| VerbForm=Part | 23 | 65 | 57 | 76 | 39 | 3 | 33 | 73 | 8 | 36 | 33 | 27 |
| Case=Ine | 84 | 15 | 24 | 4 | 1 | 89 | 5 | 87 | 85 | 35 | 24 | 35 |
| Case=Ill | 33 | 14 | 77 | 75 | 5 | 87 | 3 | 82 | 31 | 34 | 31 | 29 |
| PartForm=Past | 25 | 69 | 60 | 77 | 74 | 5 | 29 | 65 | 10 | 33 | 29 | 26 |
| Case=Ela | 83 | 62 | 68 | 67 | 4 | 19 | 4 | 86 | 42 | 32 | 42 | 24 |
| VerbForm=Inf | 13 | 31 | 21 | 49 | 84 | 8 | 84 | 12 | 86 | 77 | 84 | 25 |
| Voice=Pass | 27 | 70 | 56 | 55 | 19 | 4 | 79 | 71 | 27 | 31 | 27 | 25 |
| Person=1 | 9 | 8 | 73 | 69 | 11 | 26 | 6 | 75 | 29 | 78 | 11 | 28 |
| Case=Ade | 86 | 37 | 69 | 66 | 7 | 30 | 11 | 80 | 82 | 4 | 69 | 26 |
| Person=0 | 40 | 82 | 13 | 53 | 72 | 31 | 74 | 15 | 77 | 30 | 72 | 23 |
| Connegative=Yes | 22 | 63 | 20 | 7 | 26 | 15 | 23 | 6 | 17 | 84 | 22 | 32 |
| Polarity=Neg | 37 | 36 | 12 | 50 | 34 | 47 | 30 | 32 | 49 | 28 | 34 | 8 |
| InfForm=1 | 19 | 35 | 63 | 24 | 14 | 35 | 83 | 14 | 88 | 27 | 63 | 7 |
| Person[psor]=3 | 55 | 34 | 38 | 60 | 76 | 43 | 18 | 3 | 81 | 26 | 55 | 18 |
| PronType=Dem | 43 | 33 | 22 | 87 | 40 | 44 | 71 | 53 | 46 | 25 | 43 | 21 |
| Person=2 | 30 | 32 | 11 | 11 | 24 | 16 | 77 | 57 | 22 | 37 | 24 | 16 |
| Case=All | 54 | 75 | 81 | 65 | 12 | 36 | 13 | 79 | 2 | 23 | 13 | 22 |
| AdpType=Post | 21 | 73 | 80 | 73 | 66 | 22 | 25 | 22 | 79 | 21 | 66 | 25 |
| PartForm=Pres | 50 | 10 | 23 | 20 | 29 | 25 | 41 | 60 | 43 | 13 | 41 | 17 |
| Derivation=Minen | 31 | 3 | 83 | 83 | 22 | 29 | 15 | 7 | 19 | 20 | 22 | 28 |
| NumType=Card | 16 | 68 | 30 | 88 | 82 | 21 | 87 | 77 | 5 | 6 | 30 | 32 |
| Case=Ess | 35 | 28 | 55 | 3 | 13 | 2 | 26 | 63 | 34 | 7 | 34 | 23 |
| Case=Tra | 53 | 12 | 17 | 51 | 70 | 28 | 43 | 9 | 38 | 8 | 43 | 16 |
| Typo=Yes | 28 | 76 | 64 | 5 | 83 | 24 | 40 | 64 | 13 | 9 | 40 | 29 |
| Mood=Cnd | 77 | 67 | 19 | 23 | 36 | 53 | 39 | 20 | 26 | 10 | 36 | 21 |
| PronType=Ind | 24 | 24 | 74 | 78 | 77 | 20 | 24 | 58 | 47 | 11 | 47 | 25 |
| Mood=Imp | 34 | 23 | 8 | 8 | 25 | 17 | 75 | 16 | 23 | 5 | 25 | 6 |

Table A.2. The ranks of $\hat{\mu}_k(m)$ for each entity tag in Finnish. The morphological tags that are in the first 10 for at least one entity tag are shown. (2/3)

| Morphological Tag | ORG | TIT | PER | TIM | LOC | DATE | PRO | MISC | EVENT | OUTSIDE | median | std. dev. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| PronType=Rel | 78 | 27 | 37 | 63 | 28 | 74 | 35 | 56 | 73 | 41 | 37 | 16 |
| Number[psor]=Sing | 49 | 39 | 14 | 58 | 30 | 73 | 45 | 51 | 72 | 60 | 45 | 11 |
| AdpType=Prep | 26 | 72 | 82 | 17 | 68 | 72 | 21 | 18 | 80 | 61 | 68 | 25 |
| Derivation=Lainen | 41 | 2 | 78 | 71 | 16 | 71 | 22 | 70 | 18 | 62 | 22 | 26 |
| Degree=Sup | 38 | 58 | 66 | 70 | 31 | 70 | 19 | 61 | 25 | 63 | 31 | 4 |
| NumType=Ord | 11 | 6 | 75 | 59 | 33 | 84 | 81 | 4 | 9 | 64 | 33 | 32 |
| Clitic=Ko | 59 | 56 | 61 | 47 | 58 | 68 | 37 | 36 | 70 | 65 | 59 | 11 |
| Clitic=Kaan | 56 | 55 | 53 | 13 | 38 | 67 | 70 | 54 | 68 | 66 | 56 | 19 |
| Degree=Cmp | 82 | 54 | 26 | 81 | 46 | 66 | 46 | 44 | 67 | 67 | 46 | 12 |
| Derivation=Sti | 81 | 53 | 59 | 84 | 37 | 65 | 68 | 52 | 65 | 68 | 65 | 11 |
| Derivation=Inen | 79 | 4 | 71 | 62 | 27 | 34 | 73 | 19 | 21 | 69 | 71 | 24 |
| PronType=Int | 44 | 40 | 15 | 15 | 56 | 64 | 69 | 55 | 64 | 70 | 56 | 19 |
| Number[psor]=Plur | 52 | 64 | 65 | 28 | 75 | 63 | 44 | 21 | 62 | 71 | 62 | 20 |
| Clitic=Ka | 73 | 51 | 42 | 43 | 53 | 62 | 58 | 34 | 61 | 72 | 58 | 13 |
| Person[psor]=2 | 58 | 50 | 29 | 46 | 32 | 61 | 31 | 23 | 60 | 73 | 32 | 16 |
| Clitic=Han | 80 | 49 | 10 | 16 | 20 | 60 | 54 | 48 | 58 | 59 | 54 | 15 |
| Clitic=S | 51 | 48 | 31 | 33 | 52 | 59 | 49 | 49 | 57 | 57 | 51 | 9 |
| Mood=Pot | 39 | 13 | 32 | 32 | 67 | 58 | 27 | 10 | 56 | 40 | 39 | 17 |
| Case=Abe | 75 | 47 | 36 | 19 | 50 | 27 | 64 | 35 | 55 | 56 | 55 | 13 |
| Reflex=Yes | 76 | 46 | 49 | 48 | 45 | 57 | 66 | 46 | 54 | 39 | 54 | 5 |
| Case=Acc | 65 | 45 | 43 | 29 | 47 | 56 | 62 | 43 | 53 | 42 | 53 | 8 |
| Clitic=Pa | 72 | 44 | 47 | 41 | 65 | 55 | 65 | 39 | 52 | 43 | 65 | 5 |
| PronType=Rcp | 71 | 43 | 39 | 45 | 55 | 54 | 48 | 38 | 66 | 44 | 55 | 5 |
| Derivation=Ja | 46 | 78 | 33 | 25 | 80 | 52 | 28 | 62 | 59 | 45 | 46 | 17 |
| Derivation=Vs | 48 | 7 | 44 | 74 | 48 | 75 | 47 | 68 | 69 | 46 | 48 | 25 |
| Derivation=Ton | 42 | 25 | 28 | 26 | 35 | 39 | 32 | 37 | 35 | 47 | 35 | 8 |
| Clitic=Ko,S | 45 | 26 | 67 | 37 | 49 | 51 | 42 | 45 | 50 | 48 | 49 | 9 |
| Clitic=Pa,S | 70 | 41 | 58 | 31 | 51 | 50 | 60 | 47 | 71 | 50 | 60 | 7 |
| Case=Com | 69 | 52 | 45 | 30 | 54 | 46 | 57 | 42 | 51 | 52 | 54 | 8 |
| Derivation=Ttain | 68 | 57 | 46 | 72 | 57 | 37 | 55 | 41 | 41 | 53 | 55 | 12 |
| Clitic=Han,Ko | 66 | 59 | 50 | 42 | 59 | 49 | 51 | 40 | 44 | 54 | 51 | 7 |
| Style=Arch | 74 | 22 | 34 | 56 | 60 | 48 | 50 | 28 | 48 | 55 | 50 | 14 |
| PartForm=Neg | 64 | 29 | 40 | 39 | 61 | 38 | 52 | 33 | 45 | 74 | 52 | 16 |
| Derivation=U | 62 | 30 | 51 | 36 | 62 | 42 | 56 | 26 | 40 | 58 | 56 | 11 |
| Clitic=Han,Pa | 60 | 61 | 52 | 34 | 63 | 41 | 59 | 31 | 39 | 51 | 59 | 11 |
| Derivation=Llinen | 61 | 60 | 54 | 35 | 64 | 40 | 61 | 29 | 36 | 49 | 61 | 10 |
| Derivation=Tar | 67 | 19 | 48 | 38 | 42 | 80 | 53 | 30 | 76 | 17 | 53 | 22 |
| *UNKNOWN* | 1 | 77 | 84 | 1 | 86 | 85 | 86 | 88 | 83 | 76 | 84 | 32 |

Table A.3. The ranks of $\hat{\mu}_k(m)$ for each entity tag in Finnish. The morphological tags that are in the first 10 for at least one entity tag are shown. (3/3)

| Morphological Tag | ORG | TIT | PER | TIM | LOC | DATE | PRO | MISC | EVENT | OUTSIDE | median | std. dev. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Abbr=Yes | 7 | 5 | 1 | 22 | 78 | 82 | 85 | 74 | 4 | 82 | 7 | 32 |
| PronType=Prs | 32 | 21 | 18 | 44 | 43 | 83 | 34 | 25 | 63 | 15 | 34 | 24 |
| InfForm=3 | 57 | 20 | 27 | 14 | 71 | 81 | 36 | 17 | 20 | 16 | 36 | 25 |
| Foreign=Yes | 5 | 11 | 86 | 27 | 23 | 69 | 17 | 5 | 11 | 81 | 17 | 30 |
| Clitic=Kin | 29 | 18 | 6 | 6 | 81 | 79 | 7 | 8 | 33 | 18 | 29 | 27 |
| Case=Ins | 15 | 17 | 79 | 64 | 69 | 9 | 20 | 59 | 24 | 79 | 24 | 27 |
| PartForm=Agt | 36 | 16 | 62 | 12 | 41 | 77 | 72 | 69 | 30 | 19 | 41 | 28 |
| Case=Abl | 85 | 71 | 72 | 18 | 18 | 18 | 38 | 67 | 75 | 38 | 72 | 22 |
| Person[psor]=1 | 47 | 66 | 25 | 57 | 73 | 76 | 67 | 24 | 74 | 24 | 67 | 21 |
| InfForm=2 | 18 | 38 | 35 | 10 | 85 | 10 | 80 | 11 | 15 | 80 | 35 | 27 |

Table A.4. The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown. (1/5)

| Morphological Tag | ORG | LOC | PER | median | std. dev. |
|---|---|---|---|---|---|
| Noun | 176 | 181 | 2 | 89 | 0 |
| A3sg | 2 | 180 | 178 | 90 | 0 |
| Pnon | 181 | 179 | 181 | 181 | 0 |
| Nom | 180 | 178 | 180 | 180 | 0 |
| Verb | 8 | 147 | 173 | 90 | 0 |
| Adj | 3 | 170 | 176 | 89 | 0 |
| P3sg | 4 | 2 | 169 | 86 | 0 |
| Posˆ DB | 17 | 160 | 171 | 94 | 0 |
| Punc | 20 | 5 | 7 | 13 | 0 |
| A3pl | 7 | 10 | 162 | 84 | 0 |
| Pos | 13 | 16 | 4 | 8 | 0 |
| Prop | 1 | 1 | 1 | 1 | 0 |
| Acc | 12 | 169 | 174 | 93 | 0 |
| Zero | 50 | 173 | 166 | 108 | 0 |
| P2sg | 177 | 174 | 179 | 178 | 0 |
| Gen | 179 | 175 | 177 | 178 | 0 |
| Nomˆ DB | 9 | 166 | 170 | 89 | 0 |
| Dat | 178 | 176 | 175 | 176 | 0 |
| Loc | 19 | 177 | 161 | 90 | 0 |
| Adverb | 46 | 14 | 19 | 32 | 0 |
| Verbˆ DB | 40 | 149 | 45 | 42 | 0 |
| Conj | 28 | 6 | 18 | 23 | 0 |
| Num | 21 | 13 | 9 | 15 | 0 |
| Pass | 59 | 19 | 17 | 38 | 0 |
| P3pl | 14 | 12 | 70 | 42 | 0 |
| Adjˆ DB | 15 | 172 | 156 | 85 | 0 |
| PastPart | 47 | 53 | 40 | 43 | 0 |
| Past | 62 | 153 | 14 | 38 | 0 |
| Det | 88 | 54 | 146 | 117 | 0 |
| Inf2 | 23 | 162 | 23 | 23 | 0 |
| Postp | 64 | 21 | 152 | 108 | 0 |
| Card | 36 | 18 | 16 | 26 | 0 |
| Pron | 56 | 55 | 20 | 38 | 0 |
| Imp | 16 | 7 | 6 | 11 | 0 |
| Abl | 174 | 168 | 165 | 169 | 0 |
| PresPart | 48 | 56 | 160 | 104 | 0 |
| A2sg | 18 | 8 | 5 | 11 | 0 |
| Ness | 11 | 150 | 147 | 79 | 0 |
| Caus | 39 | 154 | 154 | 96 | 0 |
| Aorˆ DB | 32 | 17 | 164 | 98 | 0 |
| Prog1 | 92 | 57 | 65 | 78 | 0 |

Table A.5. The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown. (2/5)

| Morphological Tag | ORG | LOC | PER | median | std. dev. |
|---|---|---|---|---|---|
| P1sg | 24 | 30 | 8 | 16 | 0 |
| A1sg | 52 | 159 | 157 | 104 | 0 |
| Almost | 6 | 4 | 139 | 72 | 0 |
| Locˆ DB | 30 | 171 | 22 | 26 | 0 |
| Cardˆ DB | 31 | 22 | 24 | 27 | 0 |
| Rel | 25 | 167 | 159 | 92 | 0 |
| A1pl | 44 | 144 | 25 | 34 | 0 |
| Narr | 45 | 138 | 38 | 41 | 0 |
| Inf1 | 87 | 52 | 42 | 64 | 0 |
| Fut | 86 | 25 | 144 | 115 | 0 |
| Negˆ DB | 85 | 61 | 77 | 81 | 0 |
| Cop | 170 | 139 | 39 | 104 | 0 |
| A2pl | 41 | 28 | 11 | 26 | 0 |
| Able | 80 | 63 | 82 | 81 | 0 |
| Acquire | 71 | 50 | 36 | 53 | 0 |
| Pers | 54 | 65 | 80 | 67 | 0 |
| Opt | 51 | 29 | 13 | 32 | 0 |
| ByDoingSo | 84 | 66 | 81 | 82 | 0 |
| Causˆ DB | 83 | 67 | 74 | 78 | 0 |
| P1pl | 82 | 135 | 148 | 115 | 0 |
| Recip | 60 | 68 | 30 | 45 | 0 |
| PCAbl | 81 | 152 | 64 | 72 | 0 |
| PCDat | 79 | 69 | 145 | 112 | 0 |
| Withˆ DB | 175 | 157 | 33 | 104 | 0 |
| Cond | 43 | 31 | 35 | 39 | 0 |
| Inf3 | 55 | 70 | 51 | 53 | 0 |
| Ques | 168 | 62 | 31 | 99 | 0 |
| PresPartˆ DB | 42 | 24 | 141 | 91 | 0 |
| Nounˆ DB | 5 | 3 | 138 | 71 | 0 |
| Equ | 37 | 15 | 15 | 26 | 0 |
| Inf2ˆ DB | 76 | 51 | 60 | 68 | 0 |
| Become | 172 | 44 | 55 | 113 | 0 |
| Recipˆ DB | 53 | 134 | 21 | 37 | 0 |
| Reflex | 169 | 35 | 83 | 126 | 0 |
| While | 93 | 34 | 41 | 67 | 0 |
| Narrˆ DB | 171 | 33 | 155 | 163 | 0 |
| FitFor | 34 | 156 | 149 | 91 | 0 |
| PCAblˆ DB | 109 | 37 | 57 | 83 | 0 |
| AfterDoingSo | 149 | 36 | 34 | 91 | 0 |
| Ord | 26 | 11 | 12 | 19 | 0 |

Table A.6. The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown. (3/5)

| Morphological Tag | ORG | LOC | PER | median | std. dev. |
|---|---|---|---|---|---|
| Ly | 68 | 143 | 27 | 47 | 0 |
| EDTag | 142 | 43 | 79 | 110 | 0 |
| ETTag | 135 | 49 | 84 | 109 | 0 |
| P2pl | 140 | 46 | 95 | 117 | 0 |
| Desr | 57 | 161 | 117 | 87 | 0 |
| Without | 73 | 140 | 50 | 61 | 0 |
| Without^DB | 139 | 48 | 118 | 128 | 0 |
| Interj | 173 | 20 | 32 | 102 | 0 |
| Neces | 72 | 47 | 119 | 95 | 0 |
| PCDat^DB | 63 | 45 | 120 | 91 | 0 |
| Since | 137 | 145 | 48 | 92 | 0 |
| JustLike^DB | 136 | 71 | 121 | 128 | 0 |
| Related | 74 | 64 | 44 | 59 | 0 |
| Pass^DB | 150 | 73 | 122 | 136 | 0 |
| Real | 151 | 126 | 123 | 137 | 0 |
| When | 77 | 125 | 124 | 100 | 0 |
| Prog2 | 152 | 124 | 125 | 138 | 0 |
| Agt^DB | 70 | 123 | 126 | 98 | 0 |
| AsIf^DB | 67 | 137 | 47 | 57 | 0 |
| FeelLike | 153 | 151 | 127 | 140 | 0 |
| JustLike | 165 | 122 | 140 | 152 | 0 |
| PCIns | 164 | 121 | 130 | 147 | 0 |
| Gen^DB | 167 | 119 | 150 | 158 | 0 |
| WithoutHavingDoneSo | 163 | 112 | 128 | 145 | 0 |
| InBetween | 29 | 26 | 129 | 79 | 0 |
| NotState | 75 | 118 | 116 | 95 | 0 |
| Time | 162 | 117 | 134 | 148 | 0 |
| Rel^DB | 161 | 27 | 142 | 151 | 0 |
| DemonsP | 160 | 111 | 132 | 146 | 0 |
| PersP | 159 | 115 | 29 | 94 | 0 |
| AsLongAs | 158 | 114 | 133 | 145 | 0 |
| Ord^DB | 157 | 113 | 115 | 136 | 0 |
| PCNom^DB | 156 | 127 | 85 | 120 | 0 |
| Real^DB | 155 | 120 | 113 | 134 | 0 |
| Dim | 154 | 164 | 137 | 145 | 0 |
| Adv | 61 | 136 | 26 | 43 | 0 |
| Dup | 143 | 133 | 153 | 148 | 0 |
| Dat^DB | 134 | 132 | 46 | 90 | 0 |
| Inf | 69 | 146 | 135 | 102 | 0 |

Table A.7. The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown. (4/5)

| Morphological Tag | ORG | LOC | PER | median | std. dev. |
|---|---|---|---|---|---|
| Insˆ DB | 102 | 83 | 96 | 99 | 0 |
| ActOf | 101 | 110 | 97 | 99 | 0 |
| PCAcc | 99 | 81 | 151 | 125 | 0 |
| WithoutBeingAbleToHaveDoneSo | 98 | 80 | 98 | 98 | 0 |
| Distˆ DB | 97 | 79 | 99 | 98 | 0 |
| A3sgˆ DB | 113 | 78 | 100 | 106 | 0 |
| QuesP | 114 | 77 | 101 | 107 | 0 |
| *BLANK* | 115 | 76 | 102 | 108 | 0 |
| Hastilyˆ DB | 116 | 75 | 103 | 109 | 0 |
| Percent | 131 | 90 | 104 | 117 | 0 |
| EverSinceˆDB | 130 | 74 | 105 | 117 | 0 |
| Adamantly | 129 | 91 | 106 | 117 | 0 |
| EverSince | 128 | 93 | 107 | 117 | 0 |
| A3plˆ DB | 127 | 108 | 108 | 117 | 0 |
| Stay | 126 | 107 | 109 | 117 | 0 |
| Futˆ DB | 125 | 106 | 110 | 117 | 0 |
| Repeat | 124 | 105 | 111 | 117 | 0 |
| Prog1ˆ DB | 123 | 104 | 112 | 117 | 0 |
| Rangeˆ DB | 121 | 103 | 59 | 90 | 0 |
| NotAbleState | 118 | 102 | 63 | 90 | 0 |
| Stayˆ DB | 117 | 101 | 58 | 87 | 0 |
| Reflexˆ DB | 166 | 100 | 78 | 122 | 0 |
| ReflexP | 105 | 99 | 76 | 90 | 0 |
| ActOfˆ DB | 119 | 98 | 73 | 96 | 0 |
| Quesˆ DB | 120 | 97 | 72 | 96 | 0 |
| Whileˆ DB | 122 | 128 | 71 | 96 | 0 |
| Ratioˆ DB | 100 | 82 | 69 | 84 | 0 |
| InBetweenˆ DB | 138 | 94 | 68 | 103 | 0 |
| Prog2ˆ DB | 141 | 95 | 67 | 104 | 0 |
| Dupˆ DB | 148 | 96 | 66 | 107 | 0 |
| Distribˆ DB | 147 | 39 | 61 | 104 | 0 |
| SinceDoingSoˆ DB | 94 | 72 | 131 | 112 | 0 |

Table A.8. The ranks of $\hat{\mu}_k(m)$ for each entity tag in Turkish. The morphological tags that are in the first 20 for at least one entity tag are shown. (5/5)

| Morphological Tag | ORG | LOC | PER | median | std. dev. |
|---|---|---|---|---|---|
| Ins | 38 | 163 | 167 | 102 | 0 |
| Agt | 22 | 9 | 172 | 97 | 0 |
| Neg | 35 | 141 | 168 | 101 | 0 |
| Pres | 49 | 155 | 158 | 103 | 0 |
| With | 27 | 158 | 10 | 18 | 0 |
| Aor | 58 | 32 | 163 | 110 | 0 |
| FutPart | 91 | 148 | 143 | 117 | 0 |
| PCNom | 90 | 58 | 43 | 66 | 0 |
| Zero^DB | 33 | 59 | 37 | 35 | 0 |
| Demons | 89 | 60 | 75 | 82 | 0 |
| Able^DB | 146 | 40 | 62 | 104 | 0 |
| Quant | 145 | 38 | 53 | 99 | 0 |
| Become^DB | 65 | 41 | 54 | 59 | 0 |
| AsIf | 66 | 142 | 28 | 47 | 0 |
| *UNKNOWN* | 10 | 165 | 3 | 6 | 0 |
| Acquire^DB | 144 | 42 | 56 | 100 | 0 |
| Abl^DB | 133 | 129 | 136 | 134 | 0 |
| PCIns^DB | 132 | 131 | 86 | 109 | 0 |
| Dist | 112 | 130 | 87 | 99 | 0 |
| Related^DB | 78 | 116 | 49 | 63 | 0 |
| SinceDoingSo | 111 | 23 | 52 | 81 | 0 |
| FitFor^DB | 110 | 92 | 88 | 99 | 0 |
| PCGen | 95 | 109 | 89 | 92 | 0 |
| ESTag | 108 | 89 | 90 | 99 | 0 |
| Range | 107 | 88 | 91 | 99 | 0 |
| Hastily | 106 | 87 | 92 | 99 | 0 |
| Ratio | 104 | 86 | 93 | 98 | 0 |
| When^DB | 96 | 85 | 94 | 95 | 0 |
| Distrib | 103 | 84 | 114 | 108 | 0 |