Master's Thesis

Joint Prediction of Morphosyntactic Categories for Fine-Grained Arabic Part-of-Speech Tagging Exploiting Tag Dictionary Information

Go Inoue

March 14, 2019

Graduate School of Information Science
Nara Institute of Science and Technology

A Master's Thesis
submitted to Graduate School of Information Science,
Nara Institute of Science and Technology
in partial fulfillment of the requirements for the degree of
Master of ENGINEERING

Go Inoue

Thesis Committee:

Professor Yuji Matsumoto	(Supervisor)
Professor Satoshi Nakamura	(Co-supervisor)
Associate Professor Masashi Shimbo	(Co-supervisor)
Assistant Professor Hiroyuki Shindo	(Co-supervisor)

Joint Prediction of Morphosyntactic Categories for Fine-Grained Arabic Part-of-Speech Tagging Exploiting Tag Dictionary Information*

Go Inoue

Abstract

Part-of-speech (POS) tagging for morphologically rich languages such as Arabic is a challenging problem because of their enormous tag sets. One reason for this is that in the tagging scheme for such languages, a complete POS tag is formed by combining tags from multiple tag sets defined for each morphosyntactic category. Previous approaches in Arabic POS tagging modeled each morphosyntactic tagging task individually, without utilizing shared information between the tasks. In this work, we propose an approach that utilizes this information by jointly modeling multiple morphosyntactic tagging tasks with a multi-task learning framework. We also propose a method of incorporating tag dictionary information into our neural models by combining word representations with representations of the sets of possible tags. Our experiments showed that the joint model with tag dictionary information results in a state-of-the-art accuracy with 91.38% on the Penn Arabic Treebank data set.

Keywords:

Arabic, Part-of-Speech Tagging, Multi-task Learning

*Master's Thesis, Graduate School of Information Science, Nara Institute of Science and Technology, March 14, 2019.

アラビア語の高粒度な品詞タグ付けのための辞書情報を活用した形態統語的カテゴリの同時予測*

井上剛

内容梗概

アラビア語などの形態的に豊かな言語の品詞タグ付けは，英語など形態的に乏しい言語の品詞タグ付けに比べ，タグセットが膨大になるため，困難な問題である．これは，言語固有の情報を反映した高粒度な品詞タグが，各形態統語的カテゴリごとに定義されたタグの組み合わせによって構成されるためである．既存のアラビア語品詞タグ付けでは，各形態統語的カテゴリを個別にモデル化しており，カテゴリ間で有益な情報を共有できていなかった．本研究では，マルチタスク学習の枠組みを用いて，各形態統語的カテゴリを同時にモデル化する手法を提案する．また，入力語に対して各形態統語的カテゴリが取りうるタグを登録した辞書情報をモデルに組み込むことで，さらなる性能向上が得られることを示す．Penn Arabic Treebank を用いた評価実験の結果，提案手法は 91.38%の正解率を達成した．

キーワード

アラビア語, 品詞タグ付け, マルチタスク学習

*奈良先端科学技術大学院大学 情報科学研究科 修士論文, 2019 年 3 月 14 日.

Acknowledgments

I would like to thank my supervisor Professor Yuji Matsumoto for guiding me patiently throughout my research and writing of this thesis. I am fortunate enough to be part of such a diverse, exciting, and world-class research group as Matsumoto lab. I am deeply grateful to Professor Satoshi Nakamura for his insightful comments and suggestions on various occasions including Seminar II. I am also indebted to Associate Professor Masashi Shimbo for many valuable comments and corrections to my thesis. I would like to thank Assistant Professor Shindo Hiroyuki for teaching me how to do research from the beginning. I owe a very important debt to him for being so patient with me who had almost zero background in NLP when I started this program. I want to thank Dr. Hiroshi Noji for providing constructive comments in weekly lab meetings and research group meetings.

I would also like to especially thank Dr. Nizar Habash at New York University Abu Dhabi (NYUAD) for accepting me working as a visiting scholar in his group for a year. He has taught me how Arabic NLP is fascinating and interesting from various perspectives. The work in this thesis is greatly founded upon his extensive work on Arabic NLP for over a decade. I am also grateful to the members in the CAMeL lab at NYUAD: Salam Khalifa, Alexander Erdmann, Ossama Obeid, Nasser Zalmout, Fadhl Al Eryani, Dr. Mai Oudah, Hind Saddiki, and Dima Taji for the stimulating discussions and the fun we had during my stay in Abu Dhabi, UAE. This research stay would not have been possible without the generous support from the Tobitate! (Leap for Tomorrow) Study Abroad Initiative. I gratefully acknowledge the financial support from the Tobitate! scholarship.

Special thanks to my fellow lab mates in Computational Linguistics Laboratory at NAIST for the sleepless nights we had discussing numerous topics and the fun-time we spent together.

Last but not the least, I would like to thank my family and friends for supporting me in many ways during the Master's program.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Part-of-speech (POS) tagging is a fundamental task in natural language processing. The granularity of the POS tag set that reflects language-specific information varies from language to language. In morphologically simple languages such as English, the size of the tag set is typically less than a hundred. On the other hand, in morphologically rich languages such as Arabic, the number of theoretically possible tags can be up to 333,000 [13]. One reason for this is that in the tagging scheme for such languages, a complete POS tag is formed by combining tags from multiple tag sets defined for each morphosyntactic category. For example, a complete POS tag for the word حب *Hb*[1] 'love' can be defined as the combination of a noun from the coarse POS category, a nominative (*n*) from the case category, "not applicable" (*na*) from the mood category, and so on. The aforementioned word حب *Hb* 'love' has 23 different combinations of individual morphosyntactic tags depending on the context (Table 1.1). The enormous number of resulting tags causes fine-grained POS tagging for Arabic to be challenging.

In order to perform this task, it is beneficial to utilize information from other morphosyntactic categories when predicting a label for one category. For example, if a word is a noun, it should take one of three tags from the case category: nominative (*n*), accusative (*a*), or genitive (*g*), while it should take "not applicable" (*na*) from the mood category since mood is not defined for nominals. However, most of the previous approaches in Arabic did not utilize this information, applying one model for each task [13, 26, 28]. To make use of this information, we propose an approach that jointly models multiple morphosyntactic prediction tasks using a multi-task learning scheme. Specifically, we adopt parameter sharing in our bi-directional LSTM model in the hope

[1] We use the Buckwalter transliteration scheme [4] to represent Arabic characters.

	word	pos	gen	num	cas	mod	asp	per	vox	stt	prc3	prc2	prc1	prc0	enc0
1	حب	verb	m	s	na	i	p	3	a	na	0	0	0	0	0
2	حب	noun	m	s	u	na	na	na	na	i	0	0	0	0	0
3	حب	noun	m	s	u	na	na	na	na	c	0	0	0	0	0
4	حب	noun	m	s	u	na	na	na	na	d	0	0	0	0	0
5	حب	noun	m	s	n	na	na	na	na	c	0	0	0	0	0
6	حب	noun	m	s	n	na	na	na	na	d	0	0	0	0	0
7	حب	noun	m	s	a	na	na	na	na	c	0	0	0	0	0
8	حب	noun	m	s	a	na	na	na	na	d	0	0	0	0	0
9	حب	noun	m	s	g	na	na	na	na	c	0	0	0	0	0
10	حب	noun	m	s	g	na	na	na	na	d	0	0	0	0	0
11	حب	noun	m	s	n	na	na	na	na	i	0	0	0	0	0
12	حب	noun	m	s	g	na	na	na	na	i	0	0	0	0	0
13	حب	noun	m	s	u	na	na	na	na	i	0	0	0	0	0
14	حب	noun	m	s	u	na	na	na	na	c	0	0	0	0	0
15	حب	noun	m	s	u	na	na	na	na	d	0	0	0	0	0
16	حب	noun	m	s	n	na	na	na	na	c	0	0	0	0	0
17	حب	noun	m	s	n	na	na	na	na	d	0	0	0	0	0
18	حب	noun	m	s	a	na	na	na	na	c	0	0	0	0	0
19	حب	noun	m	s	a	na	na	na	na	d	0	0	0	0	0
20	حب	noun	m	s	g	na	na	na	na	c	0	0	0	0	0
21	حب	noun	m	s	g	na	na	na	na	d	0	0	0	0	0
22	حب	noun	m	s	n	na	na	na	na	i	0	0	0	0	0
23	حب	noun	m	s	g	na	na	na	na	i	0	0	0	0	0

Table 1.1: Possible combinations of individual morphosyntactic tags for the word حب *Hb*. The morphosyntactic categories shown in the table are: coarse POS (pos), gender (gen), number (num), case (cas), mood (mod), aspect (asp), person (per), voice (vox), state (stt), four proclitics (prc0, prc1, prc2, prc3), and one enclitic (enc).

that the shared parameters will store information beneficial to multiple tasks. To further boost the performance, we propose a method of incorporating tag dictionary information into our neural models by combining word representations with representations of the sets of possible tags.

Our experiments showed that the joint model with tag dictionary information yields the best accuracy on the Penn Arabic Treebank data set with 91.38%.

Chapter 2

Background

2.1 Arabic Language

In this section, we describe the Arabic language and the challenges it poses in the context of natural language processing (NLP) tasks. More detailed information on this topic can be found in Habash [12].

Modern Standard Arabic and Dialects

The Arabic language is classified as one of the Semitic languages in the Afro-Asiatic language family, spoken in an area from West Africa to the Arabian Gulf [9] (Figure 2.1[1]). It has multiple variants within the language where one particular variant, Modern Standard Arabic (MSA, العربية الفصحى *AlErbyp AlfSHY*), has a special status as the standard variety. MSA is primarily used in the media and education, whereas the other variants, dialects, are used in the daily communication. The Arabic dialects are often classified regionally as Maghreb (North Africa), Nile Basin (Egypt/Sudan), Levant, Gulf, and Yemen [3] as shown in the Figure 2.1. They substantially differ from MSA in terms of various linguistics aspects, including phonology, morphology, lexical choice, and syntax [12].

In this thesis, we focus on MSA, the standard variant of the Arabic language. In the following sections, we refer to MSA as Arabic unless specified otherwise.

[1] This figure was created using `https://mapchart.net/`.

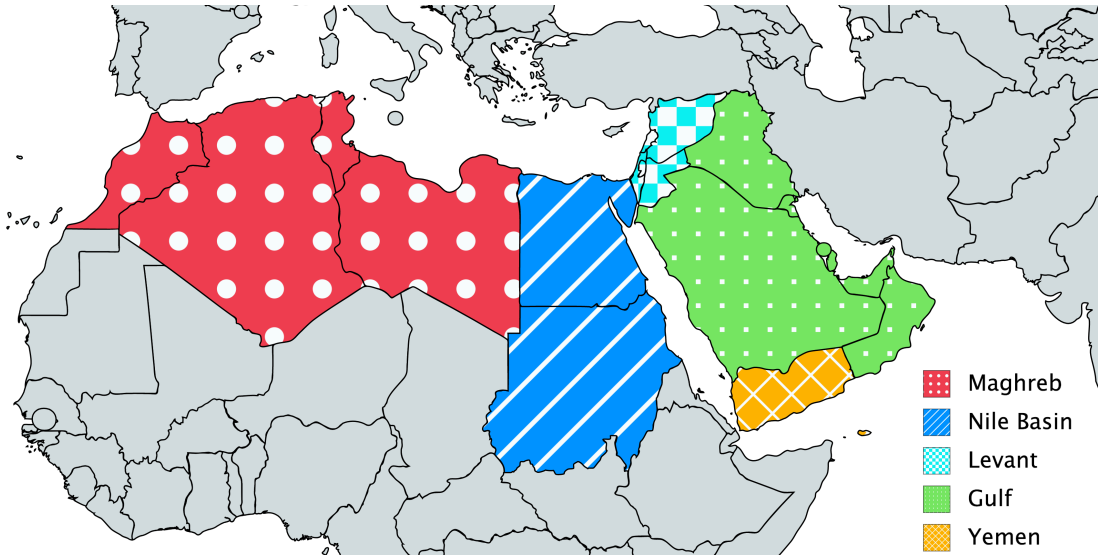

Figure 2.1: A map of regions where Arabic is spoken. The five regions correspond to the five Arabic dialects following the classification in Bouamor et al. [3]. Modern Standard Arabic (MSA) is used in all the five regions.

Arabic Orthography

Arabic is written from right to left. There are two types of symbols in the Arabic script: letters and diacritics. Arabic letters consist of 36 letters, including the basic 28 letters that correspond to Arabic's 28 consonantal sounds[2]. Diacritics are special symbols that are written above or below the letters. For example, the following word كَتَبَ *kataba* 'he wrote' is a diacritized form of the word كتب *ktb*. In this example, each diacritic symbol above the letters corresponds to the vowel sound /a/. While Arabic letters are always written, diacritics are optional. Typically, diacritics are restricted to specific genres such as religious texts or children educational texts. In the newswire genre, only 1.6% of all words have at least one diacritic indicated by their author to disambiguate the text [12]. The same sequence of letters with different diacritization can have different meaning in terms of morphological, syntactical, and lexical features. In the example above, the word كتب *ktb* can be diacritized as كَتَبَ *kataba* 'he wrote', كُتِبَ *kutiba* 'it was written', or كُتُب *kutub* 'books' depending on the context. An Arabic

[2]The letters are as follows (in alphabetical order):

ا ب ت ث ج ح خ د ذ ر ز س ش ص ض ط ظ ع غ ف ق ك ل م ن ه و ي

/a//b//t//θ//j//ħ//x//d//ð//r//z//s//ʃ//sˤ//dˤ//tˤ//ðˤ//ʕ//ɣ//f//q//k//l//m//n//h//w//j/

word can be highly ambiguous when they are undiacritized: words have an average of 12.8 different moprphosyntactic and semantic interpretations per word, most of which are associated with different diacritizations [28].

Arabic Morphology

Arabic NLP is challenging due to its complex morphology. We describe two types of important phenomena that yield this morphological complexity: cliticization and inflection [12].

Cliticization is a process by which a complex word is formed by attaching a clitic to a base word. A clitic is a bound morpheme that is syntactically independent but phonologically or orthographically dependent on the base word (cf. English contractions such as *I'm*). Arabic clitics can be divided into proclitics and enclitics depending on their position to the base word. Proclitics appear before the base word, whereas enclitics appear after the base word. Table 2.1 shows an example of a raw space-delimited word and its clitic-separated counterpart. The word in the example وسيكتبها *wsyktbhA* 'and he will write it' consists of two proclitics (و *w* 'and', س[3] *s* 'will') and one enclitic (ها *hA* 'it'), attached to the base word يكتب *yktb* 'he writes'. A space-delimited Arabic word can be considerably complex due to multiple clitics attached to the base word.

	Arabic Letter	Transliteration
space-delimited word	وسيكتبها	*wsyktbhA*
clitic-separated word	و ـ س ـ يكتب ـ ها	*w ـ s ـ yktb ـ hA*

Table 2.1: An example of a raw space-delimited word and its clitic-separated counterpart for the word وسيكتبها *wsyktbhA* 'and he will write it'. The symbol "ـ" denotes a space added after clitic separation.

Inflection is the change in the form of a word to express different grammatical categories such as person, gender, and number. Arabic verbs inflect for aspect, mood, voice, and subject (person, gender, and number). Arabic aspects can be perfective,

[3] Arabic letters have different shapes depending on their position in a word: initial, medial, final or stand-alone. The letter for the sound /s/ has the following shapes: سـ (initial), ـسـ (medial), ـس (final), and س (stand-alone).

imperfective, and imperative. Mood has three values: indicative, subjunctive, and jussive. Voice can be passive or active. The verbal subject is specified using person (1st, 2nd, or 3rd), gender (masculine or feminine), and number (single, dual, or plural). Arabic nominals (i.e., nouns, adjectives, and proper nouns) inflect for gender, number, state, and case. State has three values: definite, indefinite, and construct. The construct state is used to mark the head noun of a genitive construction. Case has three values: nominative, accusative, and genitive. Data sparsity due to the large number of inflected forms for the aforementioned grammatical categories makes NLP tasks for Arabic challenging.

2.2 Fine-grained Part-of-Speech Tagging

POS tagging takes a sequence of n words $x_{1:n}$ as input and outputs a corresponding sequence of labels $y_{1:n}$, where x_t is the t-th word in a sentence and $y_t \in T$ is the tag of x_t. In English, a POS tag is typically taken from a single tag set T. By contrast, in morphologically rich languages such as Arabic, a complete POS tag is formed by combining tags from multiple tag sets defined for each morphosyntactic category.

For example, a complete POS tag for the word حب *Hb* 'love' can be defined as the combination of a noun from the coarse POS category, a nominative (*n*) from the case category, "not applicable" (*na*) from the mood category, and so on. Formally, the fine-grained POS tag y_t^{fine} for a word x_t is defined as the conjunction of the tags $(y_t^{(1)}, y_t^{(2)}, \cdots, y_t^{(k)})$ from k tag sets $T = T^{(1)} \times T^{(2)} \times \cdots T^{(k)}$, where each $T^{(j)}$ is a tag set for a morphosyntactic category. Our purpose is then to predict all morphosyntactic categories for each word — in other words, this can be seen as a multi-class and multi-label sequential labeling problem.

In this work, we use the 14 morphosyntactic categories used in [26], a framework widely used in modern Arabic NLP tools [26, 28, 16]. The 14 categories and their possible values are shown in Table 2.2.

pos ($n = 35$)	noun, noun_num, noun_quant, noun_prop, adj, adj_comp, adj_num, adv, adv_interrog, adv_rel, pron, pron_dem, pron_exclam, pron_interrog, pron_rel, verb, verb_pseudo, part, part_dem, part_det, part_focus, part_fut, part_interrog, part_neg, part_restrict, part_verb, part_voc, prep, abbrev, punc, conj, conj_sub, interj, digit, latin
gen ($n = 3$)	m (masculine), f (feminine), na (not applicable)
num ($n = 5$)	s (singular), d (dual), p (plural), u (undefined), na
cas ($n = 5$)	n (nominative), a (accusative), g (genitive), u, na
mod ($n = 5$)	i (indicative), j (jussive), s (subjunctive), u, na
asp ($n = 4$)	i (imperfective), p (perfective), c (command), na
per ($n = 4$)	1, 2, 3, na
vox ($n = 4$)	a (active), p (passive), u, na
stt ($n = 5$)	i (indefinite), d (definite), c (constructive/poss/idafa), u, na
prc0 ($n = 10$)	0, na, Aa_prondem, AlmA_detneg, lA_neg, mA_neg, mA_part, mA_rel
prc1 ($n = 27$)	0, na,<i$_interrog, bi_part, bi_prep, bi_prog, Ea_prep, EalaY_prep, fiy_prep, hA_dem, Ha_fut, ka_prep, la_emph, la_prep, la_rc, libi_prep laHa_emphfut, laHa_rcfut, li_jus, li_prep, min_prep, sa_fut, ta_prep, wa_part, wa_prep, wA_voc, yA_voc
prc2 ($n = 9$)	0, na, fa_conj, fa_conn, fa_rc, fa_sub, wa_conj, wa_part, wa_sub
prc3 ($n = 3$)	0, na, >a_ques
enc ($n = 54$)	0, na, 1p_dobj, 1p_poss, 1p_pron, 1s_dobj, 1s_poss, 1s_pron, 2d_dobj, 2d_poss, 2d_pron, 2p_dobj, 2p_poss, 2p_pron, 2fp_dobj, 2fp_poss, 2fp_pron, 2fs_dobj, 2fs_poss, 2fs_pron, 2mp_dobj, 2mp_poss, 2mp_pron, 2ms_dobj, 2ms_poss, 2ms_pron, 3d_dobj, 3d_poss, 3d_pron, 3p_dobj, 3p_poss, 3p_pron, 3fp_dobj, 3fp_poss, 3fp_pron, 3fs_dobj, 3fs_poss, 3fs_pron, 3mp_dobj, 3mp_poss, 3mp_pron, 3ms_dobj, 3ms_poss, 3ms_pron, Ah_voc, lA_neg, ma_interrog, mA_interrog, man_interrog, man_rel, ma_rel, mA_rel, ma_sub, mA_sub

Table 2.2: The 14 morphosyntactic categories and their possible values used in Pasha et al. [26]. *n* indicates the size of the tag set.

Chapter 3

Related Work

There have been many studies on POS tagging for Arabic [7, 22, 34, 23, 26, 28, 33]. Diab et al. [7] proposed a segmentation-based approach, in which they tag each clitic-segmented token using SVMs. Mohamed and Kübler [22] proposed a word-based approach which takes space-delimited words as inputs and uses memory-based learning. Their experiment showed that the word-based approach performed better than the segmentation-based approach, avoiding segmentation error propagation. Zhang et al. [34] proposed joint modeling of segmentation, POS tagging, and dependency parsing using a randomized greedy algorithm. The aforementioned studies were focused on tagging with reduced POS tag sets whose sizes ranged from 12 to 993. However, we use one of the most fine-grained POS tag sets, with about 2,000 tags appearing in our training set.

In the context of fine-grained POS tagging, Mueller et al. [23] presented an approximated higher-order CRF for morphosyntactic tagging across six languages, assuming gold clitic segmentation. Pasha et al. [26] used an analyze-and-disambiguate approach, in which they ranked the possible analyses provided by a morphological analyzer for each space-delimited word. Shahrour et al. [28] extended their model by adjusting the outputs of Pasha et al.'s tagger by utilizing case-state classifiers that incorporate additional syntactic information provided by a dependency parser and hand-written rules.

Compared to their approaches, our model is simple but powerful: It does not assume gold clitic segmentation, since segmentation is also modeled as part of the morphosyntactic categories, nor does it require the additional pipeline process of syntactic parsing. Nonetheless, it is more accurate than the current state-of-the-art.

In parallel to our efforts, Zalmout and Habash [33] proposed a neural version of MADAMIRA [26], where they choose the correct morphological analysis from the set

of potential analyses using the outputs from the bi-LSTM classifiers for the morphological features and neural language models. They also use representations for potential POS tags obtained from a morphological dictionary and confirmed performance improvement similar to our work. They report that their approach yields slightly higher accuracy scores for the individual morphological features, but the joint features score in ours is higher.

With regard to the use of outputs from a morphological analyzer as additional features, our work is closely related to Bohnet et al. [2] and Shen et al. [29]. Bohnet et al. [2] presented a joint approach for morphological and syntactic analysis for morphologically rich languages, integrating additional features that encode whether a tag is in the dictionary or not. Shen et al. [29] proposed an approach in which they encode a sequence of possible morphosyntactic tags provided by a morphological analyzer using bi-directional LSTMs. In contrast, we provide an alternative way of encoding this information, as well as an analysis on the most influential categories in the encoded tag embeddings.

Chapter 4

Part-of-Speech Tagging Model

In this section, we first briefly describe bi-directional LSTMs. We then present our models which use bi-LSTMs for fine-grained Arabic POS tagging. We also propose a method of incorporating tag dictionary information into our neural models by combining word representations with representations of the sets of possible tags.

4.1 Bi-directional LSTMs

Recurrent neural networks (RNN) [8] are a class of neural networks that are capable of handling sequences of any length. An RNN can be seen as a function that reads the input vector x_t at time step t and calculates a hidden state $\mathbf{h}_t$ using x_t and the previous hidden state $\mathbf{h}_{t-1}$. In classification tasks, the vector $\mathbf{h}_t$ is then fed into the output layer and produces a probability distribution over the possible classes. One of the drawbacks of basic RNNs is their difficulty to train due to the so-called vanishing gradient problem. Long short term memory (LSTM) networks [15] address this issue by introducing memory cells and gate units that capture long-term dependencies.

A bi-directional LSTM network [10] is an extension of an LSTM network that allows modeling of past and future dependencies in arbitrary-length input sequences. The output vector $\mathbf{h}_t$ of a bi-LSTM is calculated by concatenating the output vector of the forward directional LSTM that reads the sequence from beginning to end with the output vector of the backward directional LSTM that reads the sequence in the reverse direction.

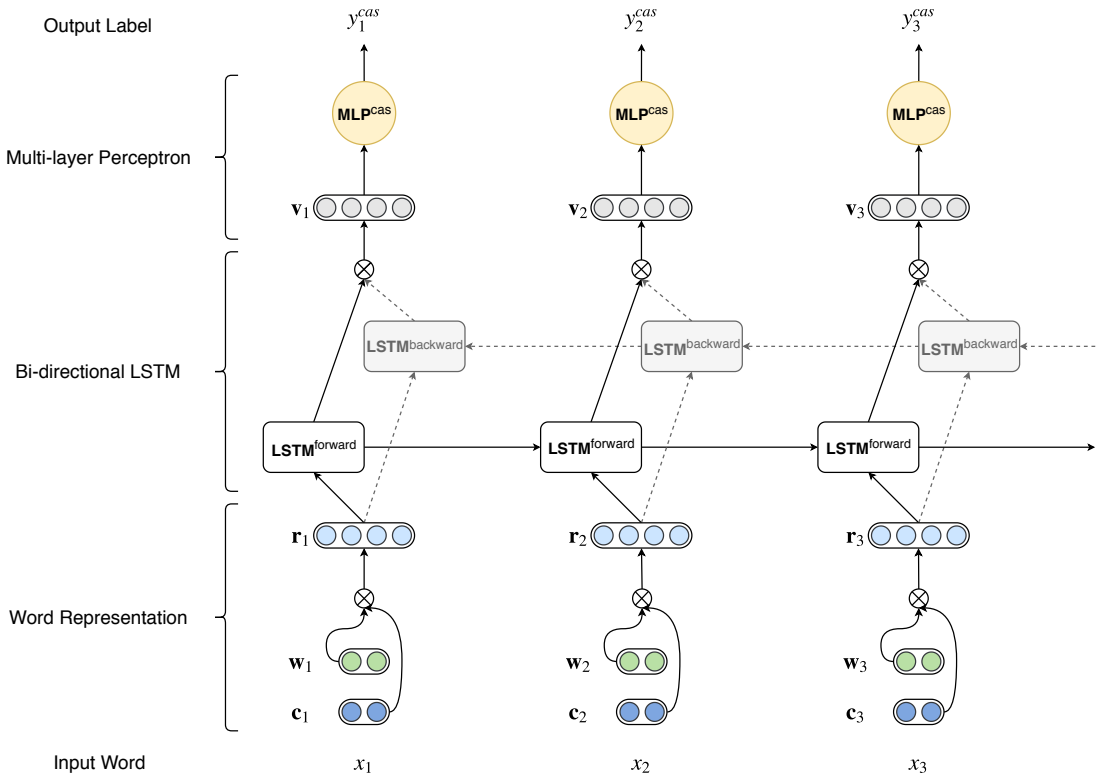

Figure 4.1: Our baseline model for the category "cas". We have one model for each category, resulting in 14 models in total. $\otimes$ is the concatenation operator.

4.2 Independent Model

For our baseline method, we use a model that independently predicts each morphosyntactic category using bi-LSTMs. Our baseline is similar to the basic model in Plank et al. [27]. Figure 4.1 illustrates an overview of our baseline model. Given a sequence of n words $x_{1:n}$, we encode each word x_t into a vector representation $\mathbf{r}_t = [\mathbf{w}_t; \mathbf{c}_t]$, which is the concatenation of the word embedding $\mathbf{w}_t$ and the character-level embedding $\mathbf{c}_t$. The character-level embedding is computed by concatenating hidden states of the character-level forward LSTM and those of the backward LSTM as depicted in Figure 4.2.

The vector representation $\mathbf{r}_t$ is then fed into the bi-LSTM model, giving the forward hidden state $\overrightarrow{\mathbf{h}}_t$ and the backward hidden state $\overleftarrow{\mathbf{h}}_t$. Both hidden states are concatenated into single vector $\mathbf{v}_t = [\overrightarrow{\mathbf{h}}_t; \overleftarrow{\mathbf{h}}_t]$ and fed into the multi-layer perceptron (MLP). Finally, we obtain the output label y_t by performing a softmax over the tag set vocabulary. We

train models separately for each morphosyntactic category, resulting in 14 models in total.

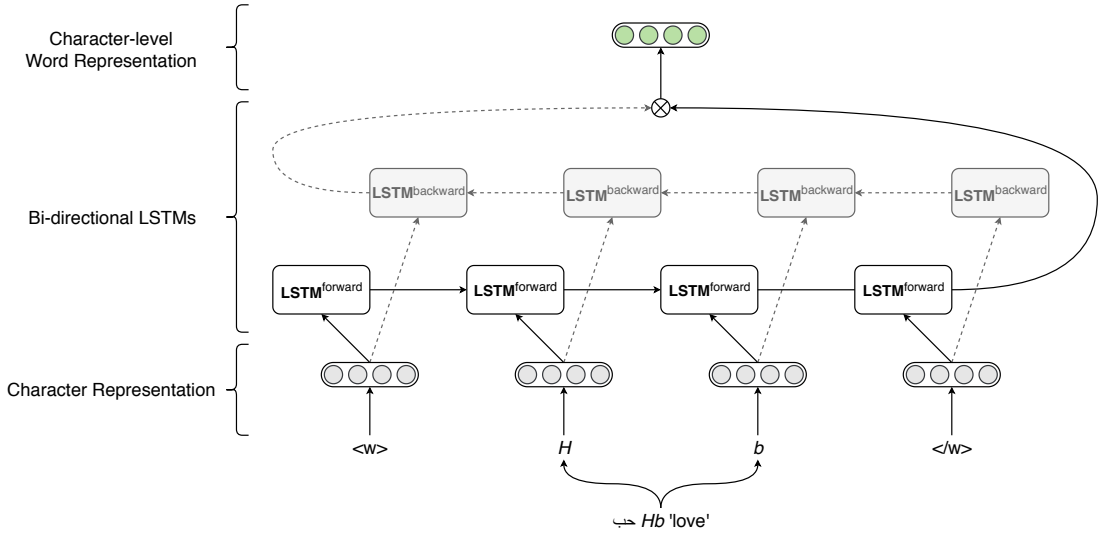

Figure 4.2: How to create character-level embeddings. <w> and </w> indicates the beginning and the end of a word.

4.3 Joint Model

Our baseline model does not share any information between morphosyntactic prediction tasks, as it is trained separately. However, it is beneficial to utilize information from other morphosyntactic categories when predicting a label for one category. In order to do this, we adopt a multi-task learning approach [5, 32, 31, 1, 21]. Specifically, we use parameter sharing in the hidden layers of the bi-LSTM model so that we can generate a unified model that can carry information beneficial to each task.

Figure 4.3 shows an overview of our joint model. The output vectors of the bi-LSTMs are fed into multiple MLPs, each performing a corresponding morphosyntactic prediction task. Our model trains to minimize the cross-entropy loss averaged across all the tasks. Let $\hat{y}^{fine}$ be the predicted fine-grained POS tag, y^{fine} be the gold fine-grained POS tag. The loss function for each input word is defined as follows:

$$L(\hat{y}^{fine}, y^{fine}) = \frac{1}{|M|} \sum_{m \in M} L(\hat{y}_m, y_m)$$

where $M = \{pos, cas, gen, ...\}$ is the set of morphosyntactic prediction tasks. $L(\hat{y}_m, y_m)$ is the cross-entropy loss for the category m, where $\hat{y}_m$ is the the predicted morphosyntactic tag for the category m, and y_m is the gold morphosyntactic tag for the category m.

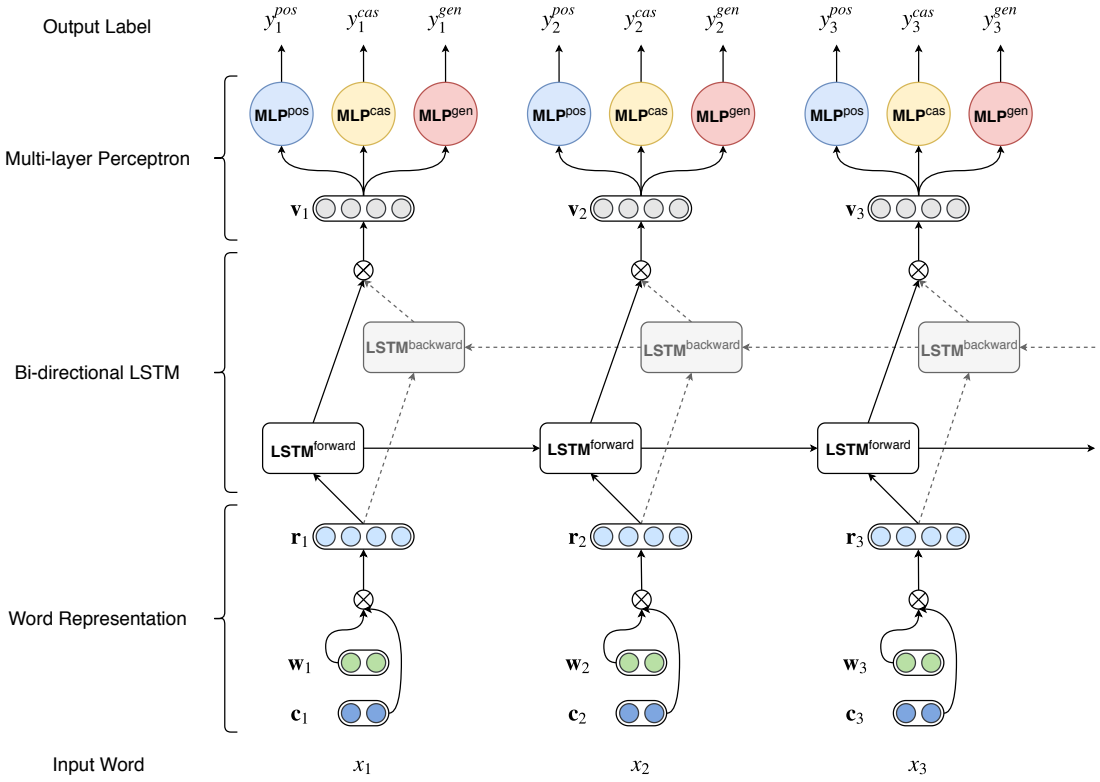

Figure 4.3: Multi-task bi-directional LSTM model for fine-grained Arabic POS tagging.

4.4 Encoding Tag Dictionary Information

One of our contributions is to incorporate tag dictionary information into our neural models by combining word representations with representations of the sets of possible tags. Unlike previous approaches that use tag dictionary information provided by a morphological analyzer as a hard constraint [13, 26, 28], we use it as a soft constraint, as well as an additional feature for our model.

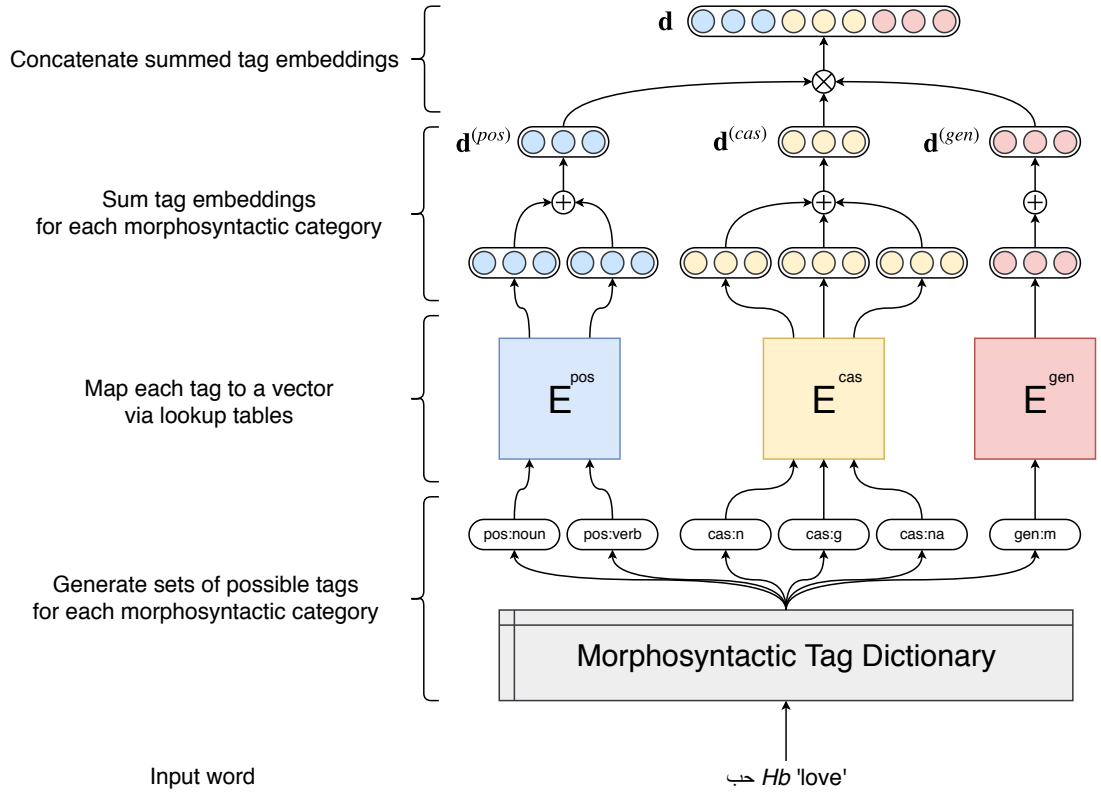

Figure 4.4: Example of how tag dictionary information is encoded. $\otimes$ is the concatenation operator and $\oplus$ is the summation operator.

The drawback of using a morphological analyzer in a pipeline fashion is that the model cannot find the correct tag in the disambiguation step if the analyzer does not return any tag candidates. Habash et al. [14] report in their error analysis that 31.3% of their tagging errors were due to this problem. To cope with this issue, we propose a method of encoding tag dictionary information into our neural models instead of using a morphological analyzer in a pipeline fashion. As such, the output of our tagger is not restricted by the output candidates that are generated by the analyzer, and our method can be applied to POS tagging with an arbitrary tag set.

Figure 4.4 illustrates how to encode tag dictionary information for the word حب *Hb* 'love.' First, the input word is given to a tag dictionary that generates sets of possible tags for each morphosyntactic category. The outputs from the dictionary are then fed into the corresponding lookup tables, giving vector representations for possible tags. For each category, we sum over the outputs from the lookup table and then concatenate

all the summed vectors into a single vector.

Formally, the encoded vector representation $\mathbf{d}_t$ for the input word x_t is computed by concatenating all the sub-vectors defined for each morphosyntactic category m:

$$\mathbf{d}_t = [\mathbf{d}_t^{(pos)}; \ldots ; \mathbf{d}_t^{(cas)}; \ldots ; \mathbf{d}_t^{(gen)}]$$

The sub-vector $\mathbf{d}_t^{(m)}$ is computed with the following equation:

$$\mathbf{d}_t^{(m)} = \sum_{d \in D_t^{(m)}} \mathbf{W}^{(m)} \mathbf{e}_d^{(m)}$$

where $D_t^{(m)}$ is the set of possible tags for the category m given the word x_t. $\mathbf{W}^{(m)} \in \mathbb{R}^{M \times K}$ is the embedding matrix for the category m, where M is the size of tag set and K is the dimension of the vector space of embedding. $\mathbf{e}_d^{(m)}$ is a one-hot vector representing the tag d for the category m. Finally, the resulting vector $\mathbf{d}_t$ is concatenated with the word embedding $\mathbf{w}_t$ and the character-level embedding $\mathbf{c}_t$, forming the input word representation $\mathbf{r}_t = [\mathbf{w}_t; \mathbf{c}_t; \mathbf{d}_t]$ for our model. Figure 4.5 illustrates the overall architecture of our proposed model.

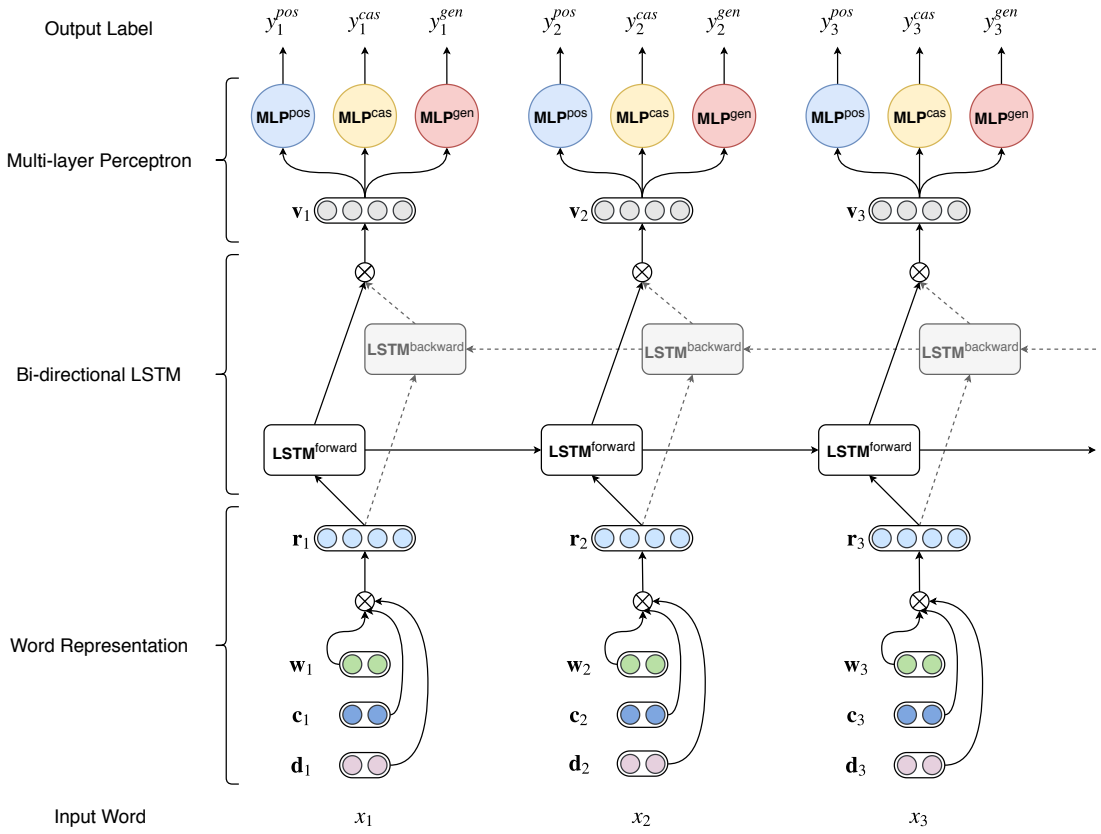

Figure 4.5: An overview of our proposed model with tag dictionary embeddings.

Chapter 5

Experiments

In this section, we present our experimental setup and results. We report tagging accuracy on two data sets: the Penn Arabic Treebank (PATB) data set and the Arabic Universal Dependencies Treebank (UD Arabic) data set. We also report the effects of tag dictionary information in both data sets.

5.1 Experimental Setup

Implementation Details

We implement all bi-LSTM models using the DyNet library [24]. We use the same hyperparameters throughout the independent and joint models, i.e., Adam with cross entropy loss, mini-batch size of a single sentence, 100 dimensions for word embeddings, 50 for character-level embeddings, 10 for each morphosyntactic dictionary embedding, 500 hidden states for each direction of bi-LSTMs, 100 hidden units for an MLP, random initialization for the embeddings, and no dropout regularization. We do not use external resources for the word embeddings in order to emulate the data availability of earlier work as much as possible. The number of epochs is optimized based on evaluation over the development set, to a maximum of 10 epochs. We use ALMOR [11], which is part of the MADAMIRA distribution [26], alongside the SAMA database [20] to create the tag dictionary.

Data Sets

The PATB Data Set

In order to compare our models with the current state-of-the-art tagger, we use the Penn Arabic Treebank (PATB, parts 1, 2 and 3) [17, 18, 19] with the same partitioning as Diab et al. [6]. The statistics of the data set are shown in Table 5.1. The data sets are pre-processed as in Pasha et al. [26] to correct annotation inconsistencies and to obtain the morphosyntactic feature representation for each word. All the Arabic characters are transliterated according to the Buckwalter transliteration scheme [4] and each numerical digit is substituted with 0.

	Train	Dev	Test
# Sentences	15,789	1,986	1,963
# Words	502,991	63,136	63,168
# Tags	2,028	1,034	1,069

Table 5.1: Number of sentences, space-delimited words, and fine-grained POS tags in the Penn Arabic Treebank data set.

The UD Arabic Data Set

In order to evaluate the performance of our models on different data in a different tagging scheme, we use the Arabic portion of Universal Dependencies Version 1.4 [25] with the provided gold tokenization. We assume gold tokenization for the sake of simplicity. The statistics of the data set are shown in Table 5.2.

	Train	Dev	Test
# Sentences	6,174	786	704
# Tokens	225,853	28,263	28,268
# Tags	327	214	213

Table 5.2: Number of sentences, tokens, and fine-grained POS tags in the UD Arabic data set.

For the fine-grained POS tag set, we use the universal POS tags and 16 of the morphological features defined in the UD Arabic data set. The annotations in the UD Arabic data set are automatically converted from the Prague Arabic Dependency Treebank [30]. Table 5.3 shows the lists of possible values for each morphosyntactic category. The annotations in UD Arabic are different from those in PATB with regard

to the choice of categories and their granularity, although there are some overlaps in categories such as gender and person. For pre-processing, each numerical digit is substituted with 0.

POS ($n = 17$)	ADJ, ADP, ADV, AUX, CONJ, DET, INTEJ, NOUN, NUM, PART, PRON, PROPN, PUNCT, SCONJ, SYM, VERB, X
Gender ($n = 3$)	Fem, Masc, EMPTY
Number ($n = 4$)	Dual, Plur, Sing, EMPTY
Case ($n = 4$)	Acc, Gen, Nom, EMPTY
Mood ($n = 5$)	Imp, Ind, Jus, Sub, EMPTY
Aspect ($n = 3$)	Imp, Perf, EMPTY
Person ($n = 4$)	1, 2, 3, EMPTY
Voice ($n = 3$)	Act, Pass, EMPTY
Definite ($n = 5$)	Com, Cons, Def, Ind, EMPTY
Abbr ($n = 2$)	Yes, EMPTY
AdpType ($n = 2$)	Prep, EMPTY
Foreign ($n = 2$)	Yes, EMPTY
Negative ($n = 2$)	Negative, EMPTY
NumForm ($n = 3$)	Digit, Word, EMPTY
NumValue ($n = 4$)	1, 2, 3, EMPTY
PronType ($n = 4$)	Dem, Prs, Rel, EMPTY
VerbForm ($n = 2$)	Fin, EMPTY

Table 5.3: The 17 morphosyntactic categories in the UD scheme (i.e., the universal POS tags and 16 morphological features) and their possible values. *n* indicates the size of the tag set.

Evaluation

Tagging Accuracy on the PATB data set

We report tagging accuracy over the 14 morphosyntactic categories and their combination, i.e., the fine-grained POS tag (**All**). For comparison, we use CamelParser [28],

the current state-of-the-art tagger. CamelParser is an improved version of the previous state-of-the-art tagger MADAMIRA [26], which ranks the possible analyses provided by a morphological analyzer using SVMs. CamelParser adjusts the outputs of MADAMIRA by utilizing case-state classifiers that incorporate additional syntactic information provided by a dependency parser and hand-written rules. The tag set used in CamelParser is compatible with the 14 morphosyntactic categories we use.

Tagging Accuracy on the UD Arabic data set
For the UD Arabic data set, we report tagging accuracy over the 17 morphosyntactic categories (i.e., the universal POS tags and 16 morphological features) and their combination (**All**). We use independent models with and without tag dictionary information and joint models with and without tag dictionary information for this data set.

5.2 Results

The PATB Data Set

Table 5.4 illustrates our experimental results on the PATB data set. The best performing model was the joint model with tag dictionary embeddings (+*Dict*), achieving an accuracy of 91.38% on the strictest metric "**All**" (i.e., the fine-grained POS tag) with an absolute improvement of 2.11% over CamelParser, the current state-of-the-art tagger. This model outperforms CamelParser in every morphosyntactic category. Among these categories, the most notable improvement is the case category (cas) with an absolute improvement of 2.08% over the current state-of-the-art system. Leaving out the dictionary embeddings (+*Dict*) reduces the performance by 1.89% absolute, but still outperforms CamelParser without using any additional resources such as a morphological analyzer or a dependency parser, indicating the effectiveness of joint modeling of morphosyntactic categories. On the other hand, the independent model gives an accuracy of 87.74%, which is 1.53% absolute worse than CamelParser. However, adding dictionary embeddings (+*Dict*) enhances the performance with an absolute improvement of 2.43% and yields the second-best accuracy, showing the impact of the additional dictionary feature. Using McNemar's test, the improvements in accuracy are all statistically significant at the 0.001 level except the joint model without the dictionary embeddings, although it is significant at the 0.1 level.

	pos	gen	num	cas	mod	asp	per
CamelParser	96.78	99.41	99.43	92.68	99.13	99.27	99.23
Independent	96.31	99.05	99.26	93.17	99.07	99.08	99.10
+*Dict*	97.07	99.33	99.51	94.70	99.31	99.34	99.35
Joint	96.24	99.27	99.16	93.48	99.18	99.19	99.20
+*Dict*	**97.21**	**99.50**	**99.59**	**94.76**	**99.41**	**99.44**	**99.47**

	vox	stt	prc0	prc1	prc2	prc3	enc	**All**
CamelParser	99.08	97.54	99.67	99.63	99.59	99.90	99.61	89.27
Independent	98.80	97.23	99.62	99.64	99.73	**99.97**	99.44	87.74
+*Dict*	99.18	98.11	99.48	99.78	**99.78**	**99.97**	99.68	90.17
Joint	98.91	97.70	99.66	99.64	99.68	**99.97**	99.58	89.49
+*Dict*	**99.25**	**98.24**	**99.71**	**99.81**	99.73	99.96	**99.71**	**91.38**

Table 5.4: Tagging accuracies on the PATB data set. **All** is the percentage where all categories were correct (i.e., the fine-grained POS tag). +*Dict* indicates the use of the tag dictionary embeddings. Best results are in boldface.

The UD Arabic Data Set

Table 5.5 illustrates our experimental results on the UD Arabic data set. The independent model gives an accuracy of 86.34% on the metric "**All**" (i.e., the fine-grained POS tag). Adding the tag dictionary embeddings (+*Dict*) improves the accuracy with an absolute improvement of 2.72%. Unlike the PATB data set, the joint model outperformed both independent models regardless of the use of the tag dictionary embeddings. The best performing model was the joint model with the tag dictionary embeddings (+*Dict*), achieving an accuracy of 91.68%. We can observe that the overall results show similar tendencies to the results on the PATB data set in spite of the different annotation schemes.

	POS	Gender	Number	Case	Mood	Aspect
Independent	95.15	97.28	96.38	93.76	99.56	99.35
+*Dict*	96.08	98.06	97.23	94.86	99.68	99.51
Joint	95.92	97.96	96.69	94.60	99.67	99.50
+*Dict*	**96.64**	**98.32**	**97.47**	**95.43**	**99.69**	**99.58**

	Person	Voice	Definite	Abbr	AdpType	Foreign
Independent	99.37	99.14	96.40	99.88	99.75	99.16
+*Dict*	99.47	99.16	97.09	**100.00**	99.84	99.58
Joint	99.45	99.21	96.67	99.99	99.85	99.47
+*Dict*	**99.59**	**99.32**	**97.35**	99.99	**99.86**	**99.66**

	Negative	NumForm	NumValue	PronType	VerbForm	**All**
Independent	99.99	99.88	99.80	99.76	99.69	86.45
+*Dict*	99.99	**99.90**	99.80	99.79	99.73	89.17
Joint	99.99	**99.90**	**99.98**	99.81	99.78	90.36
+*Dict*	99.99	99.89	**99.98**	**99.84**	**99.84**	**91.68**

Table 5.5: Tagging accuracies on the UD Arabic data set. **All** is the percentage where all categories were correct (i.e., the fine-grained POS tag). +*Dict* indicates the use of the tag dictionary embeddings.

5.3 Analysis

Case Analysis

In Figure 5.1, we show an example of improvement by joint learning of morphosyntactic features. The example sentence is extracted from the development set in the PATB data set. We show the result of independent model on the top, and that of the joint model on the bottom. In the word قتل *qtl* 'killing', both state and case categories are incorrectly tagged as *na* (not applicable) in the independent model. Given that a noun word cannot have the label *na*, this can be attributed to the inability of the model to have access to the information from other categories such as core POS. The joint model, on the other hand, correctly predicts both categories, presumably because of the model's capability of utilizing the shared information among multiple morphosyntactic tagging tasks.

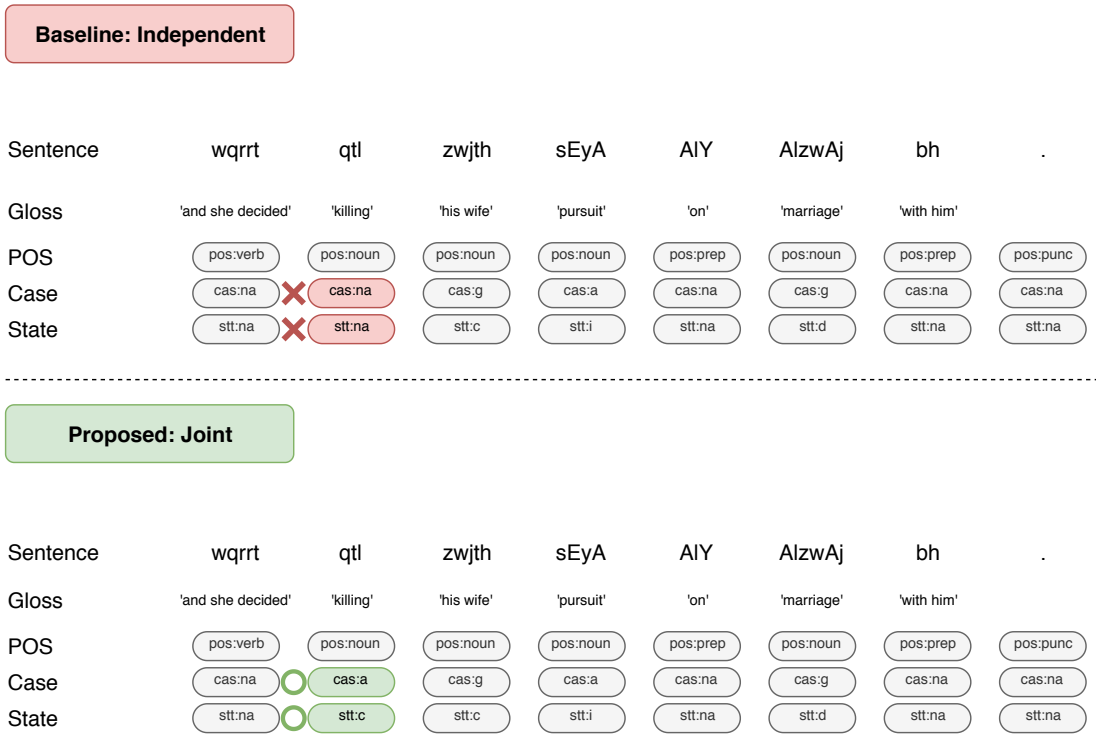

Figure 5.1: Tagging results on an example sentence extracted from the development set in the PATB data set. وقررت قتل زوجته سعيا على الزواج به. *wqrrt qtl zwjth sEyA AlzwAj bh.* 'And she decided to kill his wife to marry him.'

Most Influential Categories

For both data sets, we conduct additional experiments to investigate which morphosyntactic category in the tag dictionary embeddings contributes most to the performance. Specifically, instead of using all morphosyntactic categories to create the tag dictionary embeddings, we use only one at a time. In other words, we skip the last step of concatenating all the sub-vectors defined for each morphosyntactic category, and use only one of the sub-vectors for the tag dictionary embeddings.

The PATB Data Set

Which morphosyntactic category in the tag dictionary embeddings contributes most to the performance? Table 5.6 compares the performance of the different models, each of which uses a single morphosyntactic category in its tag dictionary embeddings. The

	pos	gen	num	cas	mod	asp	per
+*pos*	**+0.96**	**+0.25**	+0.27	**+1.00**	**+0.25**	+0.21	+0.23
+*gen*	+0.35	+0.10	+0.18	+0.34	+0.12	+0.12	+0.09
+*num*	+0.36	+0.10	**+0.43**	+0.45	+0.06	+0.07	+0.08
+*cas*	+0.51	+0.13	+0.25	+0.82	**+0.25**	+0.22	+0.23
+*mod*	+0.38	+0.10	+0.14	+0.77	+0.23	+0.23	+0.21
+*asp*	+0.47	+0.12	+0.22	+0.48	+0.22	+0.22	+0.24
+*per*	+0.26	+0.16	+0.18	+0.72	+0.24	**+0.28**	**+0.29**
+*vox*	+0.27	+0.13	+0.15	+0.65	+0.21	+0.21	+0.19
+*stt*	+0.60	+0.12	+0.20	+0.87	+0.23	+0.23	+0.22
+*prc0*	+0.31	+0.10	+0.16	+0.56	+0.06	+0.08	+0.08
+*prc1*	+0.40	+0.09	+0.21	+0.50	+0.06	-0.02	+0.06
+*prc2*	+0.23	+0.04	+0.16	+0.23	0.00	-0.01	+0.04
+*prc3*	+0.14	+0.05	+0.16	+0.33	+0.07	+0.04	+0.04
+*enc*	+0.26	+0.02	+0.12	+0.53	+0.09	+0.07	+0.07
+*all*	+0.97	+0.23	+0.43	+1.28	+0.23	+0.25	+0.27

	vox	stt	prc0	prc1	prc2	prc3	enc	**All**
+*pos*	**+0.38**	+0.46	+0.04	+0.09	+0.09	0.00	+0.08	**+1.48**
+*gen*	+0.21	+0.19	0.00	-0.06	0.00	-0.01	+0.02	+0.33
+*num*	+0.17	+0.13	+0.03	-0.02	+0.02	-0.01	+0.01	+0.63
+*cas*	+0.32	+0.41	-0.01	+0.08	+0.04	0.00	+0.06	+0.99
+*mod*	+0.31	+0.39	-0.01	+0.04	+0.05	-0.01	+0.06	+0.82
+*asp*	+0.33	+0.33	+0.02	+0.06	+0.03	0.00	+0.03	+0.68
+*per*	+0.36	+0.32	+0.01	+0.08	+0.06	0.00	+0.07	+0.78
+*vox*	+0.31	+0.29	+0.01	-0.07	-0.01	-0.01	+0.04	+0.60
+*stt*	+0.35	**+0.47**	+0.03	+0.07	+0.05	-0.01	+0.05	+0.99
+*prc0*	+0.16	+0.16	**+0.06**	+0.06	+0.05	0.00	0.00	+0.56
+*prc1*	+0.14	+0.11	+0.02	**+0.15**	+0.02	0.00	0.00	+0.69
+*prc2*	+0.12	+0.05	+0.04	-0.09	**+0.10**	-0.01	-0.02	+0.35
+*prc3*	+0.15	+0.09	+0.01	-0.05	+0.05	-0.01	+0.01	+0.28
+*enc*	+0.21	+0.22	+0.02	0.00	+0.04	-0.01	**+0.12**	+0.63
+*all*	+0.34	+0.54	+0.05	+0.17	+0.05	-0.01	+0.13	+1.89

Table 5.6: Performance comparison of the different joint models, each of which uses a single morphosyntactic category in its tag dictionary embeddings, on the PATB data set. +*m* in the leftmost column indicates the use of the category *m* to form the tag dictionary embeddings. +*all* indicates the use of all categories to form the tag dictionary embeddings. Boldfaced numbers represent the largest improvement in the category to predict (minimum of 0.05% absolute).

category that contributes most in the tag dictionary embeddings is the coarse POS category (+*pos*) with an absolute improvement of 1.48% on the metric "**All**". It is worth mentioning that case and state categories are tied for the second most contributing category, which supports CamelParser's idea that improving the prediction of case and state categories will provide further performance gains.

Looking at the effects on each category to predict, the embeddings for coarse POS (+*pos*) give the best improvement in 5 categories: coarse POS (pos), gender (gen), case (cas), mood (mod), and voice (vox). We can see that the information carried by the coarse POS category plays a central role for predicting other morphosyntactic categories, especially for the case category. On the other hand, in 8 categories, the best improvement was achieved when the category used for the tag dictionary embeddings was the same as the category to predict. The 8 categories were: coarse POS (pos), number (num), person (per), state (stt), three of the proclitics (prc0, prc1, prc2), and enclitic (enc). This result suggests that the tag dictionary embeddings of a given category behave as a soft constraint when predicting the same category, which makes intuitive sense.

The UD Arabic Data Set

Table 5.7 compares the performance of the different models, each of which uses a single morphosyntactic category in its tag dictionary embeddings, on the UD Arabic data set. As in the results on the PATB data set, the coarse POS category (+*pos*) is the category that contributes the most in the tag dictionary embeddings, giving an absolute improvement of 0.92% on the metric "**All**". It also gives the best improvement in 8 categories: POS, Aspect, Case, Definite, Foreign, Gender, Number, Person, and Voice. This result confirmed that the possible tag information from the POS category is more effective than information from the other categories.

On the other hand, unlike in the PATB data set, we do not observe a relationship between the category used for the tag dictionary embeddings and the category to predict, presumably because of the difference in the annotation schemes.

	POS	Gender	Number	Case	Mood	Aspect	Person	Voice	Definite
+*pos*	**+0.55**	**+0.30**	**+0.49**	**+0.58**	+0.04	+0.07	**+0.14**	**+0.15**	**+0.56**
+*gen*	-0.20	+0.01	-0.07	+0.05	**+0.06**	**+0.09**	+0.09	+0.13	-0.01
+*num*	+0.12	+0.06	+0.44	+0.32	+0.04	+0.01	+0.13	+0.04	+0.25
+*cas*	+0.19	+0.01	+0.33	+0.33	+0.02	+0.02	+0.08	+0.04	+0.37
+*mod*	+0.15	-0.09	+0.24	+0.26	+0.02	+0.07	+0.13	+0.14	+0.19
+*asp*	+0.19	0.00	+0.23	+0.33	-0.02	+0.06	+0.11	+0.09	+0.29
+*per*	+0.20	+0.03	+0.26	+0.38	+0.01	+0.07	+0.12	+0.07	+0.28
+*vox*	+0.08	+0.01	+0.17	+0.13	-0.01	+0.05	+0.09	+0.14	+0.21
+*stt*	+0.08	-0.06	+0.25	+0.48	-0.04	+0.04	+0.08	+0.07	+0.41
+*prc0*	-0.03	-0.06	+0.27	+0.10	+0.04	+0.02	+0.08	-0.02	+0.31
+*prc1*	-0.01	-0.01	+0.18	+0.20	+0.03	+0.02	+0.06	+0.07	+0.14
+*prc2*	-0.17	-0.12	+0.21	+0.15	+0.03	0.00	+0.01	-0.03	+0.22
+*prc3*	+0.07	-0.14	+0.29	+0.21	-0.02	+0.02	+0.08	+0.06	+0.25
+*enc*	-0.01	-0.22	+0.40	+0.10	-0.02	-0.04	-0.03	-0.05	+0.28
+*all*	+0.72	+0.36	+0.78	+0.83	+0.02	+0.08	+0.14	+0.11	+0.68

	Abbr	AdpType	Foreign	Negative	NumForm	NumValue	PronType	VerbForm	**All**
+*pos*	+0.01	-0.01	**+0.16**	0.00	+0.01	-0.02	+0.03	+0.03	**+0.92**
+*gen*	+0.01	0.00	+0.03	0.00	0.00	0.00	+0.01	+0.04	+0.08
+*num*	0.00	-0.02	+0.06	0.00	+0.02	0.00	+0.01	+0.03	+0.34
+*cas*	+0.01	0.00	+0.08	0.00	+0.02	0.00	0.00	+0.03	+0.30
+*mod*	0.00	-0.04	+0.03	0.00	0.00	-0.01	+0.01	+0.04	+0.33
+*asp*	+0.01	-0.01	+0.07	0.00	-0.01	0.00	+0.01	+0.02	+0.48
+*per*	0.00	-0.01	+0.16	0.00	+0.02	0.00	0.00	+0.02	+0.50
+*vox*	+0.01	-0.03	+0.09	0.00	-0.01	0.00	+0.01	+0.01	+0.06
+*stt*	+0.01	-0.03	+0.06	0.00	-0.01	-0.01	+0.01	+0.01	+0.28
+*prc0*	+0.01	-0.01	+0.09	0.00	0.00	-0.01	+0.01	+0.03	+0.13
+*prc1*	+0.01	-0.01	+0.12	+0.01	-0.03	-0.01	+0.01	0.00	+0.30
+*prc2*	+0.01	-0.02	+0.11	0.00	-0.02	-0.01	0.00	0.00	-0.02
+*prc3*	+0.01	-0.03	-0.02	0.00	0.00	-0.03	0.00	-0.02	-0.01
+*enc*	+0.01	-0.04	+0.03	-0.01	0.00	0.00	0.00	+0.01	-0.02
+*all*	0.00	+0.01	+0.19	0.00	-0.01	0.00	+0.03	+0.06	+1.32

Table 5.7: Performance comparison of the different joint models, each of which uses a single morphosyntactic category in its tag dictionary embeddings, on the UD Arabic data set. +*m* in the leftmost column indicates the use of the category *m* to form the tag dictionary embeddings. +*all* indicates the use of all categories to form the tag dictionary embeddings. Boldfaced numbers represent the largest improvement in the category to predict (minimum of 0.05% absolute).

Chapter 6

Conclusions

We presented an approach for fine-grained Arabic POS tagging that jointly models each morphosyntactic tagging task using a multi-task learning framework. We also proposed a method of incorporating tag dictionary information into our neural models by combining word representations with representations of the sets of possible tags. The joint model with tag dictionary information results in the best accuracy of 91.38% with an absolute improvement of 2.11% over the current state-of-the-art tagger. In addition, our experiments showed that the proposed method of encoding tag dictionary information improves the tagging accuracy even on a data set with different annotations.

One potential future direction to explore is domain adaptation to Arabic dialects, since our approach is easily applicable as it does not require construction of a morphological analyzer for each dialect. Another direction is to make use of publicly available dictionaries such as Wiktionary[1] to construct a tag dictionary. In addition, it can be worth investigating the most effective task combination in multi-task learning as in the recent studies [31, 1, 21].

[1] `https://en.wiktionary.org/`

Bibliography

[1] Bingel, J. and Søgaard, A. (2017). Identifying beneficial task relations for multi-task learning in deep neural networks. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 164–169, Valencia, Spain. Association for Computational Linguistics.

[2] Bohnet, B., Nivre, J., Boguslavsky, I., Farkas, R., Ginter, F., and Hajič, J. (2013). Joint Morphological and Syntactic Analysis for Richly Inflected Languages. *Transactions of the Association for Computational Linguistics*, 1:415–428.

[3] Bouamor, H., Habash, N., Salameh, M., Zaghouani, W., Rambow, O., Abdulrahim, D., Obeid, O., Khalifa, S., Eryani, F., Erdmann, A., and Oflazer, K. (2018). The MADAR Arabic Dialect Corpus and Lexicon. In *The International Conference on Language Resources and Evaluation*, Miyazaki, Japan.

[4] Buckwalter, T. (2002). Buckwalter Arabic Morphological Analyzer Version 1.0 LDC2002L49. Linguistic Data Consortium (LDC, Philadelphia US).

[5] Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. (2011). Natural Language Processing (Almost) from Scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

[6] Diab, M., Habash, N., Rambow, O., and Roth, R. (2013). LDC Arabic Treebanks and Associated Corpora: Data Divisions Manual. In *arXiv preprint arXiv:1309.5652*.

[7] Diab, M., Hacioglu, K., and Jurafsky, D. (2004). Automatic Tagging of Arabic Text: From Raw Text to Base Phrase Chunks. In *Proceedings of the 2004 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Short Papers*, pages 149–152, Boston, Massachusetts, USA. Association for Computational Linguistics.

[8] Elman, J. L. (1990). Finding Structure in Time. *Cognitive Science*, 14(2):179–211.

[9] Frawley, W. (2003). *International Encyclopedia of Linguistics*, volume 1. Oxford University Press Oxford ; New York, 2nd ed. edition.

[10] Graves, A. and Schmidhuber, J. (2005). Framewise Phoneme Classification with Bidirectional LSTM and Other Neural Network Architectures. *Neural Networks*, 18(5):602–610.

[11] Habash, N. (2007). Arabic Morphological Representations for Machine Translation. *Arabic Computational Morphology: Knowledge-based and Empirical Methods*, pages 263–285.

[12] Habash, N. (2010). *Introduction to Arabic Natural Language Processing*, volume 3 of *Synthesis Lectures on Human Language Technologies*. Morgan & Claypool Publishers.

[13] Habash, N. and Rambow, O. (2005). Arabic Tokenization, Part-of-Speech Tagging and Morphological Disambiguation in One Fell Swoop. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 573–580, Ann Arbor, Michigan, USA. Association for Computational Linguistics.

[14] Habash, N., Shahrour, A., and Al-Khalil, M. (2016). Exploiting Arabic Diacritization for High Quality Automatic Annotation. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 4298–4303.

[15] Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

[16] Khalifa, S., Zalmout, N., and Habash, N. (2016). YAMAMA: Yet Another Multi-Dialect Arabic Morphological Analyzer. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: System Demonstrations*, pages 223–227, Osaka, Japan. The COLING 2016 Organizing Committee.

[17] Maamouri, M., Bies, A., Kulick, S., Gaddeche, F., Mekki, W., Krouna, S., Bouziri, B., and Zaghouani, W. (2010a). Arabic Treebank: Part 1 v 4.1. Linguistic Data Consortium (LDC, Philadelphia US).

[18] Maamouri, M., Bies, A., Kulick, S., Gaddeche, F., Mekki, W., Krouna, S., Bouziri, B., and Zaghouani, W. (2011). Arabic Treebank: Part 2 v 3.1. Linguistic Data Consortium (LDC, Philadelphia US).

[19] Maamouri, M., Bies, A., Kulick, S., Krouna, S., Gaddeche, F., and Zaghouani, W. (2010b). Arabic Treebank: Part 3 v 3.2. Linguistic Data Consortium (LDC, Philadelphia US).

[20] Maamouri, M., Graff, D., Bouziri, B., Krouna, S., Bies, A., and Kulick, S. (2010c). LDC Standard Arabic Morphological Analyzer (SAMA) Version 3.1 LDC2010L01. Linguistic Data Consortium (LDC, Philadelphia US).

[21] Martínez Alonso, H. and Plank, B. (2017). When is multitask learning effective? Semantic sequence prediction under varying data conditions. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 44–53, Valencia, Spain. Association for Computational Linguistics.

[22] Mohamed, E. and Kübler, S. (2010). Is Arabic Part of Speech Tagging Feasible Without Word Segmentation? In *Proceedings of Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 705–708, Los Angeles, California, USA. Association for Computational Linguistics.

[23] Mueller, T., Schmid, H., and Schütze, H. (2013). Efficient Higher-Order CRFs for Morphological Tagging. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 322–332, Seattle, Washington, USA. Association for Computational Linguistics.

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

[25] Nivre, J., Agić, Ž., Ahrenberg, L., Aranzabe, M. J., Asahara, M., Atutxa, A., Ballesteros, M., Bauer, J., Bengoetxea, K., Berzak, Y., Bhat, R. A., Bick, E., Börstell, C., Bosco, C., Bouma, G., Bowman, S., Cebiroğlu Eryiğit, G., Celano,

G. G. A., Chalub, F., Çöltekin, Ç., Connor, M., Davidson, E., de Marneffe, M.-C., Diaz de Ilarraza, A., Dobrovoljc, K., Dozat, T., Droganova, K., Dwivedi, P., Eli, M., Erjavec, T., Farkas, R., Foster, J., Freitas, C., Gajdošová, K., Galbraith, D., Garcia, M., Gärdenfors, M., Garza, S., Ginter, F., Goenaga, I., Gojenola, K., Gökırmak, M., Goldberg, Y., Gómez Guinovart, X., Gonzáles Saavedra, B., Grioni, M., Grūzītis, N., Guillaume, B., Hajič, J., Hà Mỹ, L., Haug, D., Hladká, B., Ion, R., Irimia, E., Johannsen, A., Jørgensen, F., Kaşıkara, H., Kanayama, H., Kanerva, J., Katz, B., Kenney, J., Kotsyba, N., Krek, S., Laippala, V., Lam, L., Lê Hồng, P., Lenci, A., Ljubešić, N., Lyashevskaya, O., Lynn, T., Makazhanov, A., Manning, C., Mărănduc, C., Mareček, D., Martínez Alonso, H., Martins, A., Mašek, J., Matsumoto, Y., McDonald, R., Missilä, A., Mititelu, V., Miyao, Y., Montemagni, S., Mori, K. S., Mori, S., Moskalevskyi, B., Muischnek, K., Mustafina, N., Müürisep, K., Nguyễn Thị, L., Nguyễn Thị Minh, H., Nikolaev, V., Nurmi, H., Osenova, P., Östling, R., Øvrelid, L., Paiva, V., Pascual, E., Passarotti, M., Perez, C.-A., Petrov, S., Piitulainen, J., Plank, B., Popel, M., Pretkalniņa, L., Prokopidis, P., Puolakainen, T., Pyysalo, S., Rademaker, A., Ramasamy, L., Real, L., Rituma, L., Rosa, R., Saleh, S., Saulīte, B., Schuster, S., Seeker, W., Seraji, M., Shakurova, L., Shen, M., Silveira, N., Simi, M., Simionescu, R., Simkó, K., Šimková, M., Simov, K., Smith, A., Spadine, C., Suhr, A., Sulubacak, U., Szántó, Z., Tanaka, T., Tsarfaty, R., Tyers, F., Uematsu, S., Uria, L., van Noord, G., Varga, V., Vincze, V., Wallin, L., Wang, J. X., Washington, J. N., Wirén, M., Žabokrtský, Z., Zeldes, A., Zeman, D., and Zhu, H. (2016). Universal Dependencies 1.4. LINDAT/CLARIN digital library at the Institute of Formal and Applied Linguistics (ÚFAL), Faculty of Mathematics and Physics, Charles University.

[26] Pasha, A., Al-Badrashiny, M., Diab, M. T., El Kholy, A., Eskander, R., Habash, N., Pooleery, M., Rambow, O., and Roth, R. (2014). MADAMIRA: A Fast, Comprehensive Tool for Morphological Analysis and Disambiguation of Arabic. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, volume 14, pages 1094–1101, Reykjavik, Iceland.

[27] Plank, B., Søgaard, A., and Goldberg, Y. (2016). Multilingual Part-of-Speech Tagging with Bidirectional Long Short-Term Memory Models and Auxiliary Loss. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 412–418, Berlin, Germany. Association for Computational Linguistics.

[28] Shahrour, A., Khalifa, S., and Habash, N. (2015). Improving Arabic Diacritization through Syntactic Analysis. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1309–1315, Lisbon, Portugal. Association for Computational Linguistics.

[29] Shen, Q., Clothiaux, D., Tagtow, E., Littell, P., and Dyer, C. (2016). The Role of Context in Neural Morphological Disambiguation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 181–191, Osaka, Japan. The COLING 2016 Organizing Committee.

[30] Smrž, O., Bielicky, V., and Hajic, J. (2008). Prague Arabic Dependency Treebank: A Word on the Million Words. In *Proceedings of the Workshop on Arabic and Local Languages (LREC 2008)*, pages 16–23.

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

[32] Yang, Z., Salakhutdinov, R., and Cohen, W. (2016). Multi-Task Cross-Lingual Sequence Tagging from Scratch. *arXiv preprint arXiv:1603.06270*.

[33] Zalmout, N. and Habash, N. (2017). Don't Throw Those Morphological Analyzers Away Just Yet: Neural Morphological Disambiguation for Arabic. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 704–713, Copenhagen, Denmark. Association for Computational Linguistics.

[34] Zhang, Y., Li, C., Barzilay, R., and Darwish, K. (2015). Randomized Greedy Inference for Joint Segmentation, POS Tagging and Dependency Parsing. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 42–52, Denver, Colorado, USA. Association for Computational Linguistics.

List of Publications

Conference Papers

1. Go Inoue, Hiroyuki Shindo, and Yuji Matsumoto. 2017. "Joint Prediction of Morphosyntactic Categories for Fine-Grained Arabic Part-of-Speech Tagging Exploiting Tag Dictionary Information". In *Proceedings of the 21st Conference on Computational Natural Language Learning (CoNLL 2017)*. Vancouver, Canada, pages 421–431.

2. 井上剛, 進藤裕之, 松本裕治. 2017. アラビア語の高粒度な品詞タグ付けのための辞書情報を活用した形態統語的カテゴリの同時予測. 情報処理学会第232回自然言語処理研究会. Vol. 2017-NL-232, No.8, pages 1–9.

Other Publication

1. Go Inoue, Nizar Habash, Yuji Matsumoto, and Hiroyuki Aoyama. 2018. "A Parallel Corpus of Arabic-Japanese News Articles". In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*. Miyazaki, Japan, pages 918–924.