THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Universal Word Segmentation: Implementation and Interpretation

OPEN ACCESS

Universal Word Segmentation: Implementation and Interpretation

Yan Shao, Christian Hardmeier, Joakim Nivre
Department of Linguistics and Philology, Uppsala University
{yan.shao, christian.hardmeier, joakim.nivre}@lingfil.uu.se

Abstract

Word segmentation is a low-level NLP task that is non-trivial for a considerable number of languages. In this paper, we present a sequence tagging framework and apply it to word segmentation for a wide range of languages with different writing systems and typological characteristics. Additionally, we investigate the correlations between various typological factors and word segmentation accuracy. The experimental results indicate that segmentation accuracy is positively related to word boundary markers and negatively to the number of unique non-segmental terms. Based on the analysis, we design a small set of language-specific settings and extensively evaluate the segmentation system on the Universal Dependencies datasets. Our model obtains state-of-the-art accuracies on all the UD languages. It performs substantially better on languages that are non-trivial to segment, such as Chinese, Japanese, Arabic and Hebrew, when compared to previous work.

1 Introduction

Word segmentation is the initial step for most higher level natural language processing tasks, such as part-of-speech tagging (POS), parsing and machine translation. It can be regarded as the problem of correctly identifying word forms from a character string.

Word segmentation can be very challenging, especially for languages without explicit word boundary delimiters, such as Chinese, Japanese and Vietnamese. Even for space-delimited languages like English or Russian, relying on white space alone generally does not result in adequate segmentation as at least punctuation should usually be separated from the attached words. For some languages, the space-delimited units in the surface form are too coarse-grained and therefore often further analysed, as in the cases of Arabic and Hebrew. Even though language-specific word segmentation systems are near-perfect for some languages, it is still useful to have a single system that performs reasonably with no or minimum language-specific adaptations.

Word segmentation standards vary substantially with different definitions of the concept of a word. In this paper, we will follow the teminologies of Universal Dependencies (UD), where words are defined as basic syntactic units that do not always coincide with phonological or orthographic words. Some orthographic tokens, known in UD as multiword tokens, therefore need to be broken into smaller units that cannot always be obtained by splitting the input character sequence.[1]

To perform word segmentation in the UD framework, neither rule-based tokenisers that rely on white space nor the naive character-level sequence tagging model proposed previously (Xue, 2003) are ideal. In this paper, we present an enriched sequence labelling model for universal word segmentation. It is capable of segmenting languages in very diverse written forms. Furthermore, it simultaneously identifies the multiword tokens defined by the UD framework that cannot be resolved simply by splitting

---

[1]Note that this notion of multiword token has nothing to do with the notion of multiword expression (MWE) as discussed, for example, in Sag et al. (2002).

the input character sequence. We adapt a regular sequence tagging model, namely the bidirectional recurrent neural networks with conditional random fields (CRF) (Lafferty et al., 2001) interface as the fundamental framework (BiRNN-CRF) (Huang et al., 2015) for word segmentation.

The main contributions of this work include:

1. We propose a sequence tagging model for word segmentation, both for general purposes (mere splitting) and full UD processing (splitting plus occasional transduction).

2. We investigate the correlation between segmentation accuracy and properties of languages and writing systems, which is helpful in interpreting the gaps between segmentation accuracies across different languages as well as selecting language-specific settings for the model.

3. Our segmentation system achieves state-of-the-art accuracy on the UD datasets and improves on previous work (Straka and Straková, 2017) especially for the most challenging languages.

4. We provide an open source implementation.[2]

2 Word Segmentation in UD

The UD scheme for cross-linguistically consistent morphosyntactic annotation defines *words* as syntactic units that have a unique part-of-speech tag and enter into syntactic relations with other words (Nivre et al., 2016). For languages that use whitespace as boundary markers, there is often a mismatch between orthographic words, called *tokens* in the UD terminology, and syntactic *words*. Typical examples are clitics, like Spanish *dámelo = da me lo* (1 token, 3 words), and contractions, like French *du = de le* (1 token, 2 words). Tokens that need to split into multiple words are called *multiword tokens* and can be further subdivided into those that can be handled by simple segmentation, like English *cannot = can not*, and those that require a more complex transduction, like French *du = de le*. We call the latter *non-segmental multiword tokens*. In addition to multiword tokens, the UD scheme also allows *multitoken words*, that is, words consisting of multiple tokens, such as numerical expressions like *20 000*.

---

3 Word Segmentation and Typological Factors

We begin with the analysis of the difficulty of word segmentation. Word segmentation is fundamentally more difficult for languages like Chinese and Japanese because there are no explicit word boundary markers in the surface form (Xue, 2003). For Vietnamese, the space-segmented units are syllables that roughly correspond to Chinese characters rather than words. To characterise the challenges of word segmentation posed by different languages, we will examine several factors that vary depending on language and writing system. We will refer to these as **typological factors** although most of them are only indirectly related to the traditional notion of linguistic typology and depend more on writing system.

- **Character Set Size (CS)** is the number of unique characters, which is related to how informative the characters are to word segmentation. Each character contains relatively more information if the character set size is larger.

- **Lexicon Size (LS)** is the number of unique word forms in a dataset, which indicates how many unique word forms have to be identified by the segmentation system. Lexicon size increases as the dataset grows in size.

- **Average Word Length (AL)** is calculated by dividing the total character count by the word count. It is negatively correlated with the density of word boundaries. If the average word length is smaller, there are more word boundaries to be predicted.

- **Segmentation Frequency (SF)** denotes how likely it is that space-delimited units are further segmented. It is calculated by dividing the word count by the space-segment count. Languages like Chinese and Japanese have much higher segmentation frequencies than space-delimited languages.

- **Multiword Token Portion (MP)** is the percentage of multiword tokens that are non-segmental.

- **Multiword Token Set Size (MS)** is the number of unique non-segmental multiword tokens.

The last two factors are specific to the UD scheme but can have a significant impact on word segmentation accuracy.
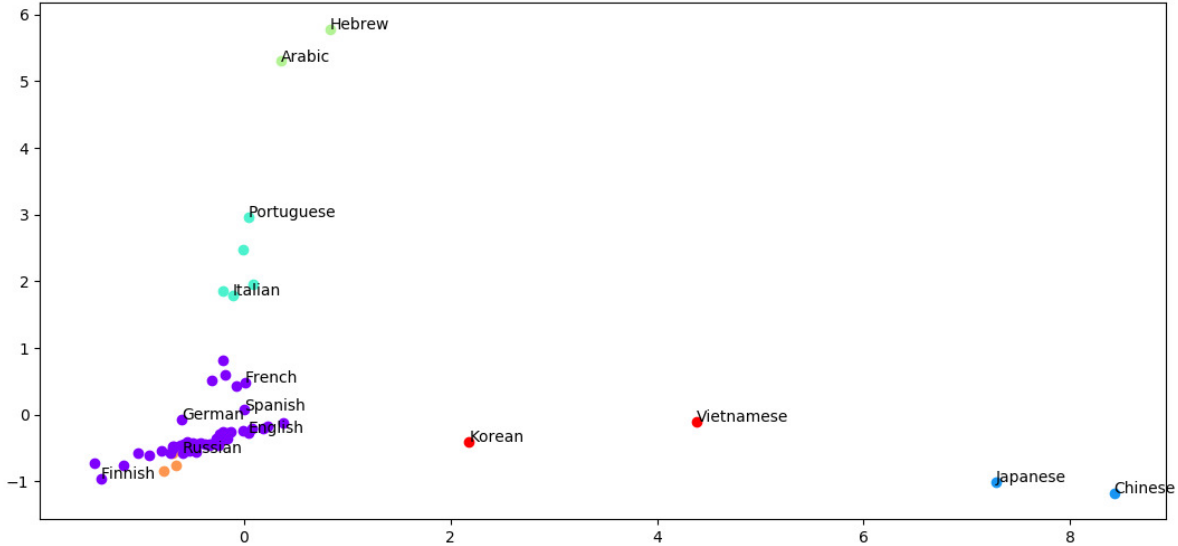
Figure 1: K-Means clustering ($K = 6$) of the UD languages. PCA is applied for dimensionality reduction.

CS	LS	AL	SF	MP	MS
0.058	0.938	0.101	-0.043	-0.060	-0.028

Table 1: Pearson product-moment correlation coefficients between dataset size and the statistical factors.

All the languages in the UD dataset are characterised and grouped by the typological factors in Figure 1. We standardise the statistics x of the proposed factors on the UD datasets with the arithmetic mean μ and the standard deviation σ as $\frac{x-\mu}{\sigma}$. We use them as features and apply K-Means clustering ($K = 6$) to group the languages. Principal component analysis (PCA) (Abdi and Williams, 2010) is used for dimensionality reduction and visualisation.

The majority of the languages in UD are space-delimited with few or no multiword tokens and they are grouped at the bottom left of Figure 1. They are statistically similar from the perspective of word segmentation. The Semitic languages Arabic and Hebrew with rich non-segmental multiword tokens are positioned at the top. In addition, languages with large character sets and high segmentation frequencies, such as Chinese, Japanese and Vietnamese are clustered together. Korean is distanced from the other space-delimited languages as it contains white-space delimiters but has a comparatively large character set. Overall, the x-axis of Figure 1 is primarily related to character set size and segmentation

Language	CS	LS	AL	SF	MP	MS
Czech	140	125,342	4.83	1.26	0.0018	9
Czech-CAC	93	66,256	5.06	1.20	0.0022	12
Czech-CLIT	96	2,774	5.30	1.14	0.0005	1
English	108	19,672	4.06	1.24	0.0	0
English-LinES	82	7,436	4.01	1.22	0.0	0
English-ParTUT	94	5,532	4.50	1.22	0.0002	6
Finnish	244	49,210	6.49	1.28	0.0	0
Finnish-FTB	95	39,717	5.94	1.14	0.0	0
French	298	42,250	4.33	1.27	0.0281	9
French-ParTUT	96	3,364	4.53	1.27	0.0344	4
French-Sequota	108	8,452	4.48	1.29	0.0277	7
Latin	57	6,927	5.05	1.28	0.0	0
Latin-ITTB	42	12,526	5.06	1.24	0.0	0
Portuguese	114	26,653	4.15	1.32	0.0746	710
Portuguese-BR	186	29,906	4.11	1.29	0.0683	35
Russian	189	25,708	5.21	1.26	0.0	0
Russian-SynTagRus	157	107,890	5.12	1.30	0.0	0
Slovenian	99	29,390	4.63	1.23	0.0	0
Slovenian-SST	40	4,534	4.29	1.12	0.0	0
Swedish	86	12,911	4.98	1.20	0.0	0
Swedish-LinES	86	9,659	4.50	1.19	0.0	0

Table 2: Different UD datasets in same languages and the statistical factors.

frequency, while the y-axis is mostly associated with multiword tokens.

Dataset sizes for different languages in UD vary substantially. Table 1 shows the correlation coefficients between the dataset size in sentence number and the six typological factors. Apart from the lexicon size, all the other factors, including multiword token set size, have no strong correlations with dataset size. From Table 2, we can see that the

| | | Char. | On considère qu'environ 50 000 Allemands du Wartheland ont péri pendant la période. |
| Tags | BEXBIIIIIIIEXBIEBIIIIIEXBIIIIEXBIIIIIIIEX $\overline{\text{B}}\overline{\text{E}}$ XBIIIIIIIIIEXBIEXBIIEXBIIIIIEXBEXBIIIIIES |

Figure 2: Tags employed for word segmentation. *50 000* is a multitoken word, while *qu'environ* and *du* are multiword tokens that should be processed differently.

factors, except for lexicon size, are relatively stable across different UD treebanks for the same language, which indicates that they do capture properties of these languages, although some variation inevitably occurs due to corpus properties like genre.

In this paper, we thoroughly investigate the correlations between the proposed statistical factors and segmentation accuracy. Moreover, we aim to find specific settings that can be applied to improve segmentation accuracy for each language group.

4 Sequence Tagging Model

Word segmentation can be modelled as a character-level sequence labelling task (Xue, 2003; Chen et al., 2015). Characters as basic input units are passed into a sequence labelling model and a sequence of tags that are associated with word boundaries are predicted. In this section, we introduce the boundary tags adopted in this paper.

Theoretically, binary classification is sufficient to indicate whether a character is the end of a word for segmentation. In practice, more fine-grained tagsets result in higher segmentation accuracy (Zhao et al., 2006). Following the work of Shao et al. (2017), we employ a baseline tagset consisting of four tags: B, I, E, and S, to indicate a character positioned at the beginning (B), inside (I), or at the end (E) of a word, or occurring as a single-character word (S).

The baseline tagset can be applied to word segmentation of Chinese and Japanese without further modification. For languages with space-delimiters, we add an extra tag X to mark the characters, mostly spaces, that do not belong to any words/tokens. As illustrated in Figure 2, the regular spaces are marked with X while the space in a multitoken word like *50 000* is disambiguated with I.

To enable the model to simultaneously identify non-segmental multiword tokens for languages like Spanish and Arabic in the UD framework, we extend the tagset by adding four tags $\overline{\text{B}}$, $\overline{\text{I}}$, $\overline{\text{E}}$, $\overline{\text{S}}$ that correspond to B, I, E, S to mark corresponding

	Tags	Applied Languages
Baseline Tags	B, I, E, S	Chinese, Japanese, ...
Boundary	X	Russian, Hindi, ...
Transduction	$\overline{\text{B}}, \overline{\text{I}}, \overline{\text{E}}, \overline{\text{S}}$	Spanish, Arabic, ...
Joint Sent. Seg.	T, U	All languages

Table 3: Tag set for universal word segmentation.

positions in non-segmental multiword tokens and to indicate their occurrences. As shown in Figure 2, the multiword token *qu'environ* is split into *qu'* and *environ* and therefore the corresponding tags are BIEBIIIIIE. This contrasts with *du*, which should be transduced into *de* and *le*. Moreover, the extra tags disambiguate whether the multiword tokens should be split or transduced according to the context. For instance, ومما *(wamimma)* in Arabic is occasionally split into و *(wa)* and مما *(mimma)* but more frequently transduced into و *(wa)*, من *(min)* and ما *(ma)* . The corresponding tags are SBIE and $\overline{\text{BIIE}}$, respectively. The transduction of the identified multiword tokens will be described in detail in the following section.

The complete tagset is summarised in Table 3. The proposed sequence model can easily be extended to perform joint sentence segmentation by adding two more tags to mark the last character of a sentence (de Lhoneux et al., 2017). T is used if the character is a single-character word and U otherwise. T and U can be used together with B, I, E, S, X for general segmentation, or with $\overline{\text{B}}$, $\overline{\text{I}}$, $\overline{\text{E}}$, $\overline{\text{S}}$ additionally for full UD processing. Joint sentence segmentation is not addressed any further in this paper.

5 Neural Networks for Segmentation

5.1 Main network

The main network for regular segmentation as well as non-segmental multiword token identification is an adaptation of the BiRNN-CRF model (Huang et al., 2015) (see Figure 3).

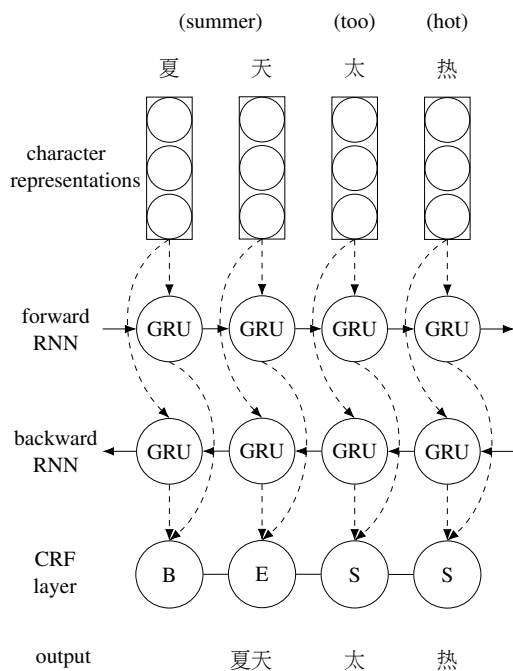The input characters can be represented as con-

Figure 3: The BiRNN-CRF model for segmentation. The dashed arrows indicate that dropout is applied.

ventional character embeddings. Alternatively, we employ the concatenated 3-gram model introduced by Shao et al. (2017). In this representation (Figure 4), the pivot character in a given context is represented as the concatenation of the character vector representation along with the local bigram and trigram vectors. The concatenated n-grams encode rich local information as the same character has different yet closely related vector representations in different contexts. For each n-gram order, we use a single vector to represent the terms that appear only once in the training set while training. These vectors are later used as the representations for unknown characters and n-grams in the development and test sets. All the embedding vectors are initialised randomly.

The character vectors are passed to the forward and backward recurrent layers. Gated recurrent units (GRU) (Cho et al., 2014) are employed as the basic recurrent cell to capture long term dependencies and sentence-level information. Dropout (Srivastava et al., 2014) is applied to both the inputs and the outputs of the bidirectional recurrent layers. A first-order chain CRF layer is added on top of the recur-
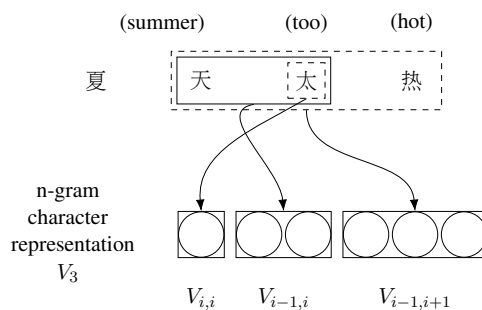

Figure 4: Concatenated 3-gram model. The third character is the pivot character in the given context.

rent layers to incorporate transition information between consecutive tags, which ensures that the optimal sequence of tags over the entire sentence is obtained. The optimal sequence can be computed efficiently via the Viterbi algorithm.

5.2 Transduction

The non-segmental multiword tokens identified by the main network are transduced into corresponding components in an additional step. Based on the statistics of the multiword tokens to be transduced on the entire UD training sets, 98.3% only have one possible transduction, which indicates that the main ambiguity of non-segmental multiword tokens comes with identification, not transduction. We therefore transduce the identified non-segmental multiword tokens in a context-free fashion. For multiword tokens with two or more valid transductions, we only adopt the most frequent one.

In most languages that have multiword tokens, the number of unique non-segmental multiword tokens is rather limited, such as in Spanish, French and Italian. For these languages, we build dictionaries from the training data to look up the multiword tokens. However, in some languages like Arabic and Hebrew, multiword tokens are very productive and therefore cannot be well covered by dictionaries generated from training data. Some of the available external dictionary resources with larger coverage, for instance the MILA lexicon (Itai and Wintner, 2008), do not follow the UD standards.

In this paper, we propose a generalising approach to processing non-segmental multiword tokens. If there are more than 200 unique multiword tokens in the training set for a language, we

Character embedding size	50
GRU/LSTM state size	200
Optimiser	Adagrad
Initial learning rate (main)	0.1
Decay rate	0.05
Gradient Clipping	5.0
Initial learning rate (encoder-decoder)	0.3
Dropout rate	0.5
Batch size	10

Table 4: Hyper-parameters for segmentation.

train an attention-based encoder-decoder (Bahdanau et al., 2015) equipped with shared long-short term memory cells (LSTM) (Hochreiter and Schmidhuber, 1997). At test time, identified non-segmental multiword tokens are first queried in the dictionary. If not found, the segmented components are generated with the encoder-decoder as character-level transduction. Overall, we utilise rich context to identify non-segmental multiword tokens, and then apply a combination of dictionary and sequence-to-sequence encoder-decoder to transduce them.

5.3 Implementation

Our universal word segmenter is implemented using the TensorFlow library (Abadi et al., 2016). Sentences with similar lengths are grouped into the same bucket and padded to the same length. We construct sub-computational graphs for each bucket so that sentences of different lengths are processed more efficiently.

Table 4 shows the hyper-parameters adopted for the neural networks. We use one set of parameters for all the experiments as we aim for a simple universal model, although fine-tuning the hyper-parameters on individual languages might result in additional improvements. The encoder-decoder is trained prior to the main network. The weights of the neural networks, including the embeddings, are initialised using the scheme introduced in Glorot and Bengio (2010). The network is trained using back-propagation. All the random embeddings are fine-tuned during training by back-propagating gradients. Adagrad (Duchi et al., 2011) with mini-batches is employed for optimization. The initial learning rate η_0 is updated with a decay rate ρ.

The encoder-decoder is trained with the unique non-segmental multiword tokens extracted from the training set. 5% of the total instances are subtracted for validation. The model is trained for 50 epochs and the score of how many outputs exactly match the references is used for selecting the weights.

For the main network, word-level F1-score is used to measure the performance of the model after each epoch on the development set. The network is trained for 30 epochs and the weight of the best epoch is selected.

To increase efficiency and reduce memory demand both for training and decoding, we truncate sentences longer than 300 characters. At decoding time, the truncated sentences are reassembled at the recorded cut-off points in a post-processing step.

6 Experiments

6.1 Datasets and Evaluation

Datasets from Universal Dependencies 2.0 (Nivre et al., 2016) are used for all the word segmentation experiments.[3] In total, there are 81 datasets in 49 languages that vary substantially in size. The training sets are available in 45 languages. We follow the standard splits of the datasets. If no development set is available, 10% of the training set is subtracted.

We adopt word-level precision, recall and F1-score as the evaluation metrics. The candidate and the reference word sequences in our experiments may not share the same underlying characters due to the transduction of non-segmental multiword tokens. The alignment between the candidate words and the references becomes unclear and therefore it is difficult to compute the associated scores. To resolve this issue, we use the longest common subsequence algorithm to align the candidate and the reference words. The matched words are compared and the evaluation scores are computed accordingly:

$$R = \frac{|c \cap r|}{|r|} \quad (1)$$

$$P = \frac{|c \cap r|}{|c|} \quad (2)$$

$$F = 2 \cdot \frac{R \cdot P}{R + P} \quad (3)$$

where c and r denote the sequences of candidate words and reference words, and $|c|$, $|r|$ are their

---

[3]We employ the version that was used in the CoNLL 2017 shared task on UD parsing.

Basic Unit	F1-score	Training Time (s)
Latin Character	82.79	572
Space-delimited Unit	87.62	218

Table 5: Different segmentation units employed for word segmentation on Vietnamese. Concatenated 3-gram is not used.
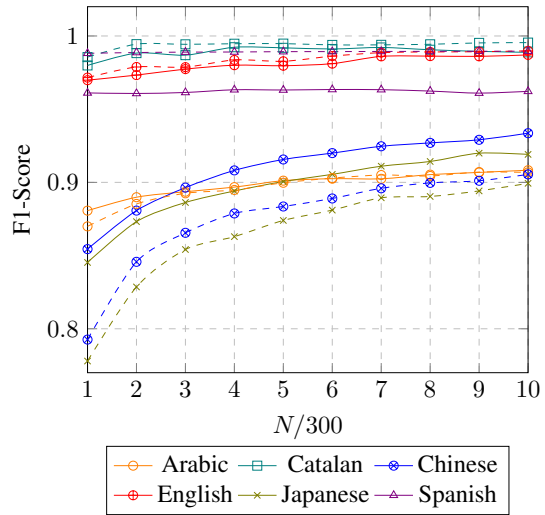

Figure 5: Segmentation results with unigram character embeddings (dashed) and concatenated 3-gram vectors for character representations with different numbers of training instances N.

lengths. $|c \cap r|$ is the number of candidate words that are aligned to reference words by the longest common subsequence algorithm. The word-level evaluation metrics adopted in this paper are different from the boundary-based alternatives (Palmer and Burger, 1997).

We adapt the evaluation script from the CoNLL 2017 shared task (Zeman et al., 2017) to calculate the scores. In the following experiments, we only report the F1-score.

In the following sections, we thoroughly investigate correlations between several language-specific characteristics and segmentation accuracy. All the experimental results in Section 6.2 are obtained on the development sets. The test sets are reserved for final evaluation, reported in Section 6.3.

6.2 Language-Specific Characteristics

6.2.1 Word-Internal Spaces

For Vietnamese and other languages with similar historical backgrounds, such as Zhuang and Hmongic languages (Zhou, 1991), the space-delimited syllables containing no punctuation are never segmented but joined into words with word-internal spaces instead. The space-delimited units can therefore be applied as the basic elements for tag prediction if we pre-split punctuation. Word segmentation for these languages thus becomes practically the same as for Chinese and Japanese.

Table 5 shows that a substantial improvement can be achieved if we use space-delimited syllables as the basic elements for word segmentation for Vietnamese. It also drastically increases both training and decoding speed as the sequence of tags to be predicted becomes much shorter.

6.2.2 Character Representation

We apply regular character embeddings and concatenated 3-gram vectors introduced in Section 5.1 to the input characters and test their performances

respectively. First, the experiments are extensively conducted on all the languages with the full training sets. The results show that the concatenated 3-gram model is substantially better than the regular character embeddings on Chinese, Japanese and Vietnamese, but notably worse on Spanish and Catalan. For all the other languages, the differences are marginal.

To gain more insights, we select six languages, namely Arabic, Catalan, Chinese, Japanese, English and Spanish for more detailed analysis via learning curve experiments. The training sets are gradually extended by 300 sentences at a time. The results are shown in Figure 5. Regardless of the amounts of training data and the other typological factors, concatenated 3-grams are better on Chinese and Japanese and worse on Spanish and Catalan. We expect the concatenated 3-gram representation to outperform simple character embeddings on all languages with a large character set but no space delimiters.

Since adopting the concatenated 3-gram model drastically enlarges the embedding space, in the following experiments, including the final testing phase, concatenated 3-grams are only applied to Chinese, Japanese and Vietnamese.
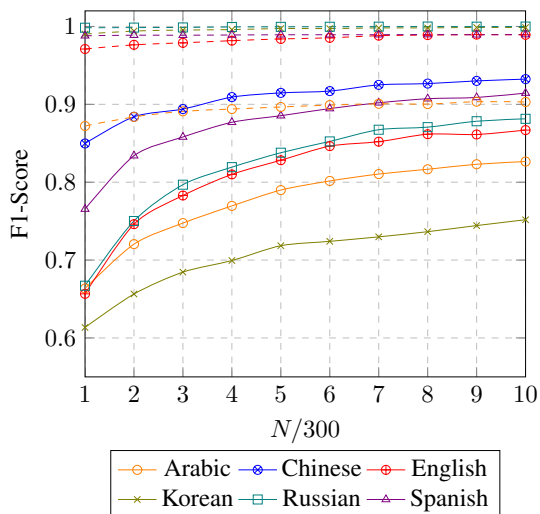
Figure 6: Segmentation results with (dashed) and without space delimiters with different numbers of training instances N.

Figure 7: Segmentation results with and without (dashed) processing non-segmental multiword tokens with different training instances N.

6.2.3 Space Delimiters

Chinese and Japanese are not delimited by spaces. Additionally, continuous writing without spaces (scriptio continua) is evidenced in most Classical Greek and Latin manuscripts. We perform two sets of learning curve experiments to investigate the impact of white space on word segmentation. In the first set, we keep the datasets in their original forms. In the second set, we omit all white space. The experimental results are presented in Figure 6.

In general, there are huge discrepancies between the accuracies with and without spaces, showing that white space acts crucially as a word boundary indicator. Retaining the original forms of the space-delimited languages, very high accuracies can be achieved even with small amounts of training data as the model quickly learns that space is a reliable word boundary indicator. Moreover, we obtain relatively lower scores on space-delimited languages when space is ignored than Chinese using comparable amounts of training data, which shows that Chinese characters are more informative to word boundary prediction, due to the large character set size.

6.2.4 Non-Segmental Multiword Tokens

The concept of multiword tokens is specific to UD. To explore how the non-segmental multiword tokens, as opposed to pure segmentation, influence
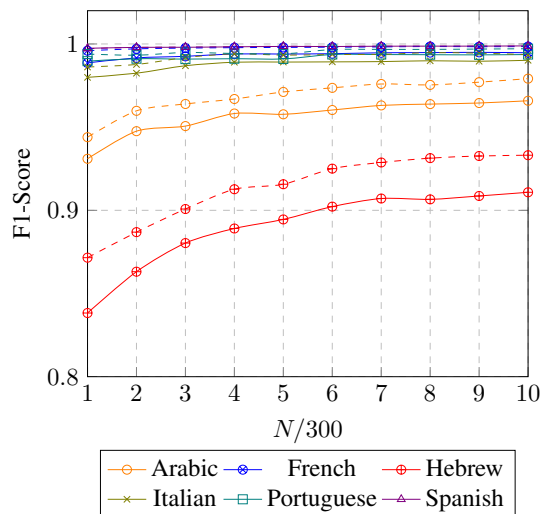
Language	Data size		Evaluation Scores	
	Training	Validation	ACC	MFS
Arabic	3,500	184	77.84	82.64
Hebrew	2,995	157	84.81	92.35

Table 6: Accuracy of the seq2seq transducer on Arabic and Hebrew.

segmentation accuracy, we conduct relevant experiments on selected languages. Similarly to the previous section, two sets of learning curve experiments are performed. In the second set, all the multiword tokens that require transduction are regarded as single words without being processed. The results are presented in Figure 7.

Word segmentation with full UD processing is notably more challenging for Arabic and Hebrew. Table 6 shows the evaluation of the encoder-decoder as the transducer for non-segmental multiword tokens on Arabic and Hebrew. The evaluation metrics ACC and MF-score (MFS) are adapted from the metrics used for machine transliteration evaluation (Li et al., 2009). ACC is exact match and MFS is based on edit distance. The transducer yields relatively higher scores on Hebrew while it is more challenging to process Arabic.

In addition, different approaches to transducing the non-segmental multiword tokens are evaluated in Table 7. In the condition None, the identified non-

428

	None	Dictionary	Transducer	Mix
Arabic	94.11	96.74	96.54	**97.27**
Hebrew	87.17	91.33	88.46	**91.85**

Table 7: Segmentation accuracies on Arabic and Hebrew with different ways of transducing non-segmental multiword tokens.

segmental multiword tokens remain unprocessed. In Dictionary, they are mapped via the dictionary derived from training data if found in the dictionary. In Transducer, they are all transduced by the attention-based encoder-decoder. In Mix, in addition to utilising the mapping dictionary, the non-segmental terms not found in the dictionary are transduced with the encoder-decoder. The results show that when the encoder-decoder is applied alone, it is worse than only using the dictionaries, but additional improvements can be obtained by combining both of them.

The accuracy differences associated with non-segmental multiword tokens are nonetheless marginal on the other languages as shown in Figure 7. Regardless of their frequent occurrences, multiword tokens are easy to process in general when the set of unique non-segmental multiword tokens is small.

6.2.5 Correlations with Accuracy

We investigate the correlations between the proposed typological factors in Section 3 and segmentation accuracy using linear regression with Huber loss (Huber, 1964). The factors are used in addition to training set size as the features to predict the segmentation accuracies in F1-score. To collect more data samples, apart from experimenting with the full training data for each set, we also use smaller sets of 500, 1,000 and 2,000 training instances to train the models respectively if the training set is large enough. The features are standardised with the arithmetic mean and the standard deviation before fitting the linear regression model.

The correlation coefficients of the linear regression model are presented in Figure 8. We can see that segmentation frequency and multiword token set size are negatively correlated with segmentation accuracy. Overall, the UD datasets are strongly biased towards space-delimited languages. Training set size is therefore not a strong factor as high accu-
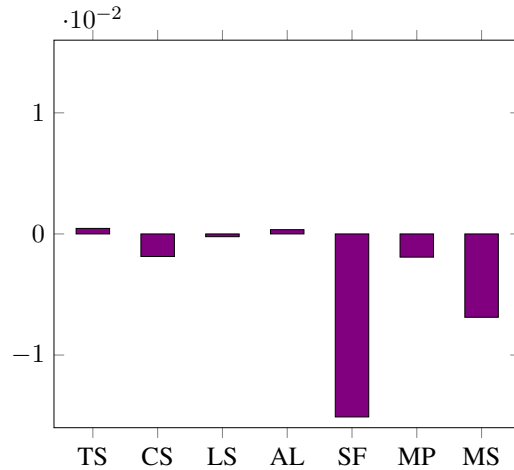

Figure 8: Correlation coefficients between segmentation accuracy and the typological factors in the linear regression model. The factors are training set size (TS), character set size (CS), lexicon size (LS), average word length (AL), segmentation frequency (SF), multitoken word portion (MP) and multitoken word size (MS).

racies can be obtained with small amounts of training data, which is consistent with the results of all the learning curve experiments. The other typological factors such as average word length and lexicon size are less relevant to segmentation accuracy. Referring back to Figure 1, segmentation frequency and multiword token set size as the most influential factors, are also the primary principal components that categorise the UD languages into different groups.

6.2.6 Language-Specific Settings

Our model obtains competitive results with only a minimal number of straightforward language-specific settings. Based on the previous analysis of segmentation accuracy and typological factors, referring back to Figure 1, we apply the following settings, targeting on specific language groups, to the segmentation system on the final test sets. The language-specific settings can be applied to new languages beyond the UD datasets based on an analysis of the typological factors.

1. For languages with word-internal spaces like Vietnamese, we first separate punctuation and then use space-delimited syllables for bound-

Space	NLTK	UDPipe	This Paper
80.86	95.64	99.47	99.45

Table 8: Average evaluation scores on UD languages, excluding Chinese, Japanese, Vietnamese, Arabic and Hebrew.

ary prediction.

2. For languages with large character sets and no space delimiters, like Chinese and Japanese, we use concatenated 3-gram representations.

3. For languages with more than 200 unique non-segmental multiword tokens, like Arabic and Hebrew, we use the encoder-decoder model for transduction.

4. For other languages, the universal model is sufficient without any specific adaptation.

6.3 Final Results

We compare our segmentation model to UDPipe (Straka and Straková, 2017) on the test sets. UDPipe contains word segmentation, POS tagging, morphological analysis and dependency parsing models in a pipeline. The word segmentation model in UDPipe is also based on RNN with GRU. For efficiency, UDPipe has a smaller character embedding size and no CRF interface. It also relies heavily on whitespace and uses specific configurations for languages in which word-internal spaces are allowed. Automatically generated suffix rules are applied jointly with a dictionary query to handle multiword tokens. Moreover, UDPipe uses language-specific hyperparameters for Chinese and Japanese.

We employ UDPipe 1.2 with the publicly available UD 2.0 models.[4] The *presegmented* option is enabled as we assume the input text to be presegmented into sentences so that only word segmentation is evaluated. In addition, the CoNLL shared task involved some test sets for which no specific training data were available. This included a number of parallel test sets of known languages, for which we apply the models trained on the standard treebanks, as well as four surprise languages, namely Buryat, Kurmanji, North Sami and Upper Sorbian, for which we use the small annotated data samples provided in addition to the test sets by the shared

---

[4]http://hdl.handle.net/11234/1-2364

task to build models and evaluation on those languages.

The main evaluation results are shown in Table 9. We also report the Macro Average F1-scores. The scores of the surprise languages are excluded and presented separately as no corresponding UDPipe models are available.

Our system obtains higher segmentation accuracy overall. It achieves substantially better accuracies on languages that are challenging to segment, namely Chinese, Japanese, Vietnamese, Arabic and Hebrew. The two systems yield very similar scores, when these languages are excluded as shown in Table 8, in which the two systems are also compared with two rule-based baselines, a simple space-based tokeniser and the tokenisation model for English in NLTK (Loper and Bird, 2002). The NLTK model obtains relatively high accuracy while the space-based baseline substantially underperforms, which indicates that relying on white space alone is insufficient for word segmentation in general. On the majority of the space-delimited languages without productive non-segmental multiword tokens, both UDPipe and our segmentation system yield near-perfect scores in Table 9. In general, referring back to Figure 1, languages that are clustered at the bottom-left corner are relatively trivial to segment.

The evaluation scores are notably lower on Semitic languages as well as languages without word delimiters. Nonetheless, our system obtains substantially higher scores on the languages that are more challenging to process.

For Chinese, Japanese and Vietnamese, our system benefits substantially from the concatenated 3-gram character representation, which has been demonstrated in Section 6.2.2. Besides, we employ a more fine-grained tagset with CRF loss instead of the binary tags adopted in UDPipe. As presented in Zhao et al. (2006), more fine-grained tagging schemes outperform binary tags, which is supported by the experimental results on morpheme segmentation reported in Ruokolainen et al. (2013).

We further investigate the merits of the fine-grained tags over the binary tags as well as the effectiveness of the CRF interface by the experiments presented in Table 10 with the variances of our segmentation system. The fine-grained tags denote the boundary tags introduced in Table 3. The binary

Dataset	UDPipe	This Paper	Dataset	UDPipe	This Paper	Dataset	UDPipe	This Paper
Ancient Greek	99.98	99.96	Ancient Greek-PROIEL	99.99	100.0	Arabic	93.77	97.16
Arabic-PUD	90.92	95.93	Basque	99.97	100.0	Bulgarian	99.96	99.93
Catalan	99.98	99.80	Chinese	90.47	93.82	Croatian	99.88	99.95
Czech	99.94	99.97	Czech-CAC	99.96	99.93	Czech-CLTT	99.58	99.64
Czech-PUD	99.34	99.62	Danish	99.83	100.0	Dutch	99.84	99.92
Dutch-LassySmall	99.91	99.96	English	99.05	99.13	English-LinES	99.90	99.95
English-PUD	99.69	99.71	English-ParTUT	99.60	99.51	Estonian	99.90	99.88
Finnish	99.57	99.74	Finnish-FTB	99.95	99.99	Finnish-PUD	99.64	99.39
French	98.81	99.39	French-PUD	98.84	97.23	French-ParTUT	98.97	99.32
French-Sequoia	99.11	99.48	Galician	99.94	99.97	Galician-TreeGal	98.66	98.07
German	99.58	99.64	German-PUD	97.94	97.74	Gothic	100.0	100.0
Greek	99.94	99.86	Hebrew	85.16	91.01	Hindi	100.0	100.0
Hindi-PUD	98.26	98.82	Hungarian	99.79	99.93	Indonesian	100.0	100.0
Irish	99.38	99.85	Italian	99.83	99.54	Italian-PUD	99.21	98.78
Japanese	92.03	93.77	Japanese-PUD	93.67	94.17	Kazakh	94.17	94.21
Korean	99.73	99.95	Latin	99.99	100.0	Latin-ITTB	99.94	100.0
Latin-PROIEL	99.90	100.0	Latvian	99.16	99.56	Norwegian-Bokmaal	99.83	99.89
Norwegian-Nynorsk	99.91	99.97	Old Church Slavonic	100.0	100.0	Persian	99.65	99.62
Polish	99.90	99.93	Portuguese	99.59	99.10	Portuguese-BR	99.85	99.52
Portuguese-PUD	99.40	98.98	Romanian	99.68	99.74	Russian	99.66	99.96
Russian-PUD	97.09	97.28	Russian-SynTagRus	99.64	99.65	Slovak	100.0	99.98
Slovenian	99.93	100.0	Slovenian-SST	99.91	100.0	Spanish	99.75	99.85
Spanish-AnCora	99.94	99.93	Spanish-PUD	99.44	99.39	Swedish	99.79	99.97
Swedish-LinES	99.93	99.98	Swedish-PUD	98.36	99.26	Turkish	98.09	97.85
Turkish-PUD	96.99	96.68	Ukrainian	99.81	99.76	Urdu	100.0	100.0
Uyghur	99.85	97.86	Vietnamese	85.53	87.79	**Average**	98.63	**98.90**

Table 9: Evaluation results on the UD test sets in F1-scores. The datasets are represented in the corresponding treebank codes. *PUD* suffix indicates the parallel test data. Two shades of green/red are used for better visualisation, with brighter colours for larger differences. Green represents that our system is better than UDPipe and red is used otherwise.

	BT	BT+CRF	FT	FT+CRF
Chinese	90.54	90.66	90.73	**91.28**
Japanese	91.54	91.64	91.88	**91.94**
Vietnamese	87.63	**87.95**	87.61	87.75
Arabic	94.47	96.74	94.73	**97.16**
Hebrew	85.34	90.74	85.53	**91.98**

Table 10: Comparison between the binary tags (BT) and the fine-grained tags (FT) as well as the effectiveness of the CRF interface on the development sets.

	Arabic	French	German	Hebrew
UDPipe	79.34	**98.91**	94.21	71.87
Our model	**91.35**	97.50	94.21	**86.17**

Table 11: Percentages of the correctly processed multiword tokens on the development sets.

tags include two basic tags `B`, `I` plus the corresponding tags $\overline{B}$, $\overline{I}$ for non-segmental multiword tokens. White space is marked as `I` instead of `X`. The concatenated 3-grams are not applied. In general, the experimental results confirm that the fine-grained tags are more beneficial except for Vietnamese. The fine-grained tagset contains more structured positional information that can be exploited by the word segmentation model. Additionally, the CRF interface leads to notable improvements, especially

for Arabic and Hebrew. The combination of the fine-grained tags with the CRF interface achieves substantial improvements over the basic binary tag model that is analogous to UDPipe.

For Arabic and Hebrew, apart from greatly benefiting from the fine-grained tagset and the CRF interface, our model is better at handling non-segmental multiword tokens (Table 11). The attention-based encoder-decoder as the transducer is much more powerful in processing the non-segmental multiword tokens that are not covered by the dictionary than the suffix rules for analysing multiword tokens in UDPipe.

UDPipe obtains higher scores on a few datasets. Our model overfits the small training data of Uyghur

| | Segmentation Accuracy | | UDPipe parser | | | | Dozat et al. (2017) | | | |
| | | | UAS | | LAS | | UAS | | LAS | |
	UDPipe	This Paper	UDPipe	This Paper	UDPipe	This Paper	UDPipe	This Paper	UDPipe	This Paper
Arabic	93.77	97.16	72.34	78.22	66.41	71.79	77.52	83.55	72.89	78.42
Chinese	90.47	93.82	63.20	67.91	59.07	63.31	71.24	76.33	68.20	73.04
Hebrew	85.16	91.01	62.14	71.18	57.82	66.59	67.61	76.39	64.02	72.37
Japanese	92.03	93.77	78.08	81.77	76.73	80.83	80.21	83.79	79.44	82.99
Vietnamese	85.53	87.79	47.72	50.87	43.10	46.03	50.28	53.78	45.54	48.86

Table 12: Extrinsic evaluations with dependency parsing on the test sets. The parsing accuracies are reported in unlabelled attachment score (UAS) and labelled attachment score (LAS).

	Space	NLTK	Sample	Transfer
Buryat	71.99	97.99	88.07	97.99 (Russian)
Kurmanji	78.97	**97.37**	93.37	96.71 (Spanish)
North Sami	79.07	99.20	92.82	**99.81** (German)
Upper Sorbian	72.35	**94.60**	93.34	93.66 (Spanish)

Table 13: Evaluation on the surprise languages.

as it yields 100.0 F1-score on the development set. For a few parallel test sets, there are punctuation marks not found in the training data that cannot be correctly analysed by our system as it is fully data-driven without any heuristic rules for unknown characters.

The evaluation results on the surprise languages are presented in Table 13. In addition to the segmentation models proposed in this paper, we present the evaluation scores of a space-based tokeniser as well as the NLTK model for English. As shown by the previous learning curve experiments in Section 6.2, very high accuracies can be obtained on the space-delimited languages with only small amounts of training data. However, in case of extreme data sparseness (less than 20 training sentences), such as for the four surprise languages in Table 13 and Kazakh in Table 9, the segmentation results are drastically lower even though the surprise languages are all space-delimited.

For the surprise languages, we find that applying segmentation models trained on a different language with more training data yields better results than relying on the small annotated samples of the target language. Considering that the segmentation model is fully character-based, we simply select the model of the language that shares the most characters with the surprise language as its segmentation model. No annotated data of the surprise language are used for model selection. As shown in Table 13, the transfer

approach achieves comparable segmentation accuracies to NLTK. For space-delimited languages with insufficient training data, it may be beneficial to employ a well-designed rule-based word segmenter as NLTK occasionally outperforms the data-driven approach.

As a form of extrinsic evaluation, we test the segmenter in a dependency parsing setup on the datasets where we obtained substantial improvements over UDPipe. We present results for the transition-based parsing model in UDPipe 1.2 and for the graph-based parser by Dozat et al. (2017). The experimental results are shown in Table 12. We can see that word segmentation accuracy has a great impact on parsing accuracy as the segmentation errors propagate. Having a more accurate word segmentation model is very beneficial for achieving higher parsing accuracy.

7 Related Work

The BiRNN-CRF model is proposed by Huang et al. (2015) and has been applied to a number of sequence labelling tasks, such as part-of-speech tagging, chunking and named entity recognition.

Our universal word segmenter is a major extension of the joint word segmentation and POS tagging system described by Shao et al. (2017). The original model is specifically developed for Chinese and only applicable to Chinese and Japanese. Apart from being language-independent, the proposed model in this paper employs an extended tagset and a complementary sequence transduction component to fully process non-segmental multiword tokens that are present in a substantial amount of languages, such as Arabic and Hebrew in particular. It is a generalised segmentation and transduction framework.

Our universal model is compared with the

	This Paper	Shao	Che	Björkelund
Chinese	93.82	95.21	91.19	92.81
Japanese	93.77	94.79	92.95	91.68
Arabic	97.16	–	93.71	95.53
Hebrew	91.01	–	85.16	91.37

Table 14: Comparison between the universal model and the language-specific models.

language-specific model of Shao et al. (2017) in Table 14. With pretrained character embeddings, ensemble decoding and joint POS tags prediction as introduced in Shao et al. (2017), considerable improvements over the universal model presented in this paper can be obtained. However, the joint POS tagging system is difficult to generalise as single characters in space-delimited languages are usually not informative for POS tagging. Additionally, compared to Chinese, sentences in space-delimited languages have a much greater number of characters on average. Combining the POS tags with segmentation tags drastically enlarges the search space and therefore the model becomes extremely inefficient both for training and tagging. The joint POS tagging model is nonetheless applicable to Japanese and Vietnamese.

Monroe et al. (2014) present a data-driven word segmentation system for Arabic based on a sequence labelling framework. An extended tagset is designed for Arabic-specific orthographic rules and applied together with hand-crafted features in a CRF framework. It obtains 98.23 F1-score on newswire Arabic Treebank,[5] 97.61 on Broadcast News Treebank,[6] and 92.10 on the Egyptian Arabic dataset.[7] For Hebrew, Goldberg and Elhadad (2013) perform word segmentation jointly with syntactic disambiguation using lattice parsing. Each lattice arc corresponds to a word and its corresponding POS tag, and a path through the lattice corresponds to a specific word segmentation and POS tagging of the sentence. The proposed model is evaluated on the Hebrew Treebank (Guthmann et al., 2009). The joint word segmentation and parsing F1-score (76.95) is reported and compared against the parsing score (85.70) with gold word segmentation. The evaluation scores re-

ported in both Monroe et al. (2014) and Goldberg and Elhadad (2013) are not directly comparable to the evaluation scores on Arabic and Hebrew in this paper, as they are obtained on different datasets.

For universal word segmentation, apart from UDPipe described in Section 6.3, there are several systems that are developed for specific language groups. Che et al. (2017) build a similar Bi-LSTM word segmentation model targeting languages without space delimiters like Chinese and Japanese. The proposed model incorporates rich statistics-based features gathered from large-scale unlabelled data, such as character unigram embeddings, character bigram embeddings and the point-wise mutual information of adjacent characters. Björkelund et al. (2017) use a CRF-based tagger for multiword token rich languages like Arabic and Hebrew. A predicted Levenshtein edit script is employed to transform the multiword tokens into their components. The evaluation scores on a selected set of languages reported in Che et al. (2017) and Björkelund et al. (2017) are included in Table 14 as well.

More et al. (2018) adapt existing morphological analysers for Arabic, Hebrew and Turkish and present ambiguous word segmentation possibilities for these languages in a lattice format (CoNLL-UL) that is compatible with UD. The CoNLL-UL datasets can be applied as external resources for processing non-segmental multiword tokens.[8]

8   Conclusion

We propose a sequence tagging model and apply it to universal word segmentation. BiRNN-CRF is adopted as the fundamental segmentation framework that is complemented by an attention-based sequence-to-sequence transducer for non-segmental multiword tokens. We propose six typological factors to characterise the difficulty of word segmentation cross different languages. The experimental results show that segmentation accuracy is primarily correlated with segmentation frequency as well as the set of non-segmental multiword tokens. Using whitespace as delimiters is crucial to word segmentation, even if the correlation between orthographic tokens and words is not perfect. For space-delimited

---

[5]LDC2010T13, LDC2011T09, LDC2010T08
[6]LDC2012T07
[7]LDC2012E93,98,89,99,107,125, LDC2013E12,21

[8]CoNLL-UL is not evaluated in our experiments as it is very recent work.

languages, very high accuracy can be obtained even with relatively small training sets, while more training data is required for high segmentation accuracy for languages without spaces. Based on the analysis, we apply a minimal number of language-specific settings to substantially improve the segmentation accuracy for languages that are fundamentally more difficult to process.

The segmenter is extensively evaluated on the UD datasets in various languages and compared with UDPipe. Apart from obtaining nearly perfect segmentation on most of the space-delimited languages, our system achieves high accuracies on languages without space delimiters such as Chinese and Japanese as well as Semitic languages with abundant multiword tokens like Arabic and Hebrew.

Acknowledgments

References

Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. 2016. TensorFlow: A system for large-scale machine learning. In *Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, pages 265–283.

Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley Interdisciplinary Reviews: Computational Statistics*, 2(4):433–459.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *International Conference on Learning Representations*.

Anders Björkelund, Agnieszka Falenska, Xiang Yu, and Jonas Kuhn. 2017. IMS at the CoNLL 2017 UD shared task: CRFs and perceptrons meet neural networks. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 40–51.

Wanxiang Che, Jiang Guo, Yuxuan Wang, Bo Zheng, Huaipeng Zhao, Yang Liu, Dechuan Teng, and Ting Liu. 2017. The HIT-SCIR system for end-to-end parsing of universal dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 52–62.

Xinchi Chen, Xipeng Qiu, Chenxi Zhu, Pengfei Liu, and Xuanjing Huang. 2015. Long short-term memory neural networks for Chinese word segmentation. In *Conference on Empirical Methods in Natural Language Processing*, pages 1197–1206.

Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *arXiv preprint arXiv:1409.1259*.

Miryam de Lhoneux, Yan Shao, Ali Basirat, Eliyahu Kiperwasser, Sara Stymne, Yoav Goldberg, and Joakim Nivre. 2017. From raw text to Universal Dependencies – look, no tags! In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies.*, pages 207–217.

Timothy Dozat, Peng Qi, and Christopher D. Manning. 2017. Stanford's graph-based neural dependency parser at the CoNLL 2017 shared task. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 20–30, Vancouver, Canada, August. Association for Computational Linguistics.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121–2159.

Xavier Glorot and Yoshua Bengio. 2010. Understanding the difficulty of training deep feedforward neural networks. In *International Conference on Artificial Intelligence and Statistics*, pages 249–256.

Yoav Goldberg and Michael Elhadad. 2013. Word segmentation, unknown-word resolution, and morphological agreement in a Hebrew parsing system. *Computational Linguistics*, 39(1):121–160, March.

Noemie Guthmann, Yuval Krymolowski, Adi Milea, and Yoad Winter. 2009. Automatic annotation of morphosyntactic dependencies in a modern Hebrew. In *Proceedings of the 1st Workshop on Treebanks and Linguistic Theories*, pages 1–12.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

Zhiheng Huang, Wei Xu, and Kai Yu. 2015. Bidirectional LSTM-CRF models for sequence tagging. *arXiv preprint arXiv:1508.01991*.

Peter J Huber. 1964. Robust estimation of a location parameter. *The annals of mathematical statistics*, pages 73–101.

Alon Itai and Shuly Wintner. 2008. Language resources for Hebrew. *Language Resources and Evaluation*, 42(1):75–98.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, ICML '01, pages 282–289.

Haizhou Li, A Kumaran, Vladimir Pervouchine, and Min Zhang. 2009. Report of NEWS 2009 machine transliteration shared task. In *Proceedings of the 2009 Named Entities Workshop: Shared Task on Transliteration*, pages 1–18.

Edward Loper and Steven Bird. 2002. NLTK: The natural language toolkit. In *Proceedings of the ACL-02 Workshop on Effective tools and methodologies for teaching natural language processing and computational linguistics-Volume 1*, pages 63–70. Association for Computational Linguistics.

Will Monroe, Spence Green, and Christopher D Manning. 2014. Word segmentation of informal arabic with domain adaptation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 206–211.

Amir More, Özlem Çetinoğlu, Çağrı Çöltekin, Nizar Habash, Benoît Sagot, Djamé Seddah, Dima Taji, and Reut Tsarfaty. 2018. CoNLL-UL: Universal morphological lattices for Universal Dependency parsing. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation*.

Joakim Nivre, Marie-Catherine de Marneffe, Filip Ginter, Yoav Goldberg, Jan Hajic, Christopher D. Manning, Ryan McDonald, Slav Petrov, Sampo Pyysalo, Natalia Silveira, Reut Tsarfaty, and Daniel Zeman. 2016. Universal dependencies v1: A multilingual treebank collection. In *Proceedings of the 10th International Conference on Language Resources and Evaluation*, pages 1659–1666.

David Palmer and John Burger. 1997. Chinese word segmentation and information retrieval. In *AAAI Spring Symposium on Cross-Language Text and Speech Retrieval*, pages 175–178.

Teemu Ruokolainen, Oskar Kohonen, Sami Virpioja, and Mikko Kurimo. 2013. Supervised morphological segmentation in a low-resource learning setting using conditional random fields. In *Proceedings of the Seventeenth Conference on Computational Natural Language Learning*, pages 29–37, Sofia, Bulgaria. Association for Computational Linguistics.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for NLP. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15. Springer.

Yan Shao, Christian Hardmeier, Jörg Tiedemann, and Joakim Nivre. 2017. Character-based joint segmentation and POS tagging for Chinese using bidirectional RNN-CRF. In *Proceedings the 8th International Joint Conference on Natural Language Processing*, pages 173–183.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Milan Straka and Jana Straková. 2017. Tokenizing, POS tagging, lemmatizing and parsing UD 2.0 with UDPipe. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 88–99.

Nianwen Xue. 2003. Chinese word segmentation as character tagging. *Computational Linguistics and Chinese Language Processing*, pages 29–48.

Daniel Zeman, Martin Popel, Milan Straka, Jan Hajic, Joakim Nivre, Filip Ginter, Juhani Luotolahti, Sampo Pyysalo, Slav Petrov, Martin Potthast, Francis Tyers, Elena Badmaeva, Memduh Gokirmak, Anna Nedoluzhko, Silvie Cinkova, Jan Hajic jr., Jaroslava Hlavacova, Václava Kettnerová, Zdenka Uresova, Jenna Kanerva, Stina Ojala, Anna Missilä, Christopher D. Manning, Sebastian Schuster, Siva Reddy, Dima Taji, Nizar Habash, Herman Leung, Marie-Catherine de Marneffe, Manuela Sanguinetti, Maria Simi, Hiroshi Kanayama, Valeria dePaiva, Kira Droganova, Héctor Martínez Alonso, Çağr Çöltekin, Umut Sulubacak, Hans Uszkoreit, Vivien Macketanz, Aljoscha Burchardt, Kim Harris, Katrin Marheinecke, Georg Rehm, Tolga Kayadelen, Mohammed Attia, Ali Elkahky, Zhuoran Yu, Emily Pitler, Saran Lertpradit, Michael Mandl, Jesse Kirchner, Hector Fernandez Alcalde, Jana Strnadová, Esha Banerjee, Ruli Manurung, Antonio Stella, Atsuko Shimada, Sookyoung Kwak, Gustavo Mendonca, Tatiana Lando, Rattima Nitisaroj, and Josie Li. 2017. CoNLL 2017 shared task: Multilingual parsing from raw text to Universal Dependencies. In *Proceedings of the CoNLL 2017 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 1–19.

Hai Zhao, Chang-Ning Huang, Mu Li, and Bao-Liang Lu. 2006. Effective tag set selection in Chinese word segmentation via conditional random field modeling. In *Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation*, pages 87–94.

Youguang Zhou. 1991. The family of Chinese character-type scripts. *Sino-Platonic Papers, 28*.