A Deep Learning System for Automatic Extraction of Typological Linguistic Information from Descriptive Grammars

Shafqat Mumtaz Virk
Språkbanken Text, Dept. of Swedish
University of Gothenburg
405 30 Gothenburg, Sweden
virk.shafqat@gmail.com

Daniel Foster
University of Gothenburg
405 30 Gothenburg, Sweden
daniel.a.foster@gmail.com

Azam Sheikh Muhammad
Chalmers University of Technology
412 96 Gothenburg, Sweden
sheikhazam@gmail.com

Raheela Saleem
Gift University
Gujranwala, Pakistan
raheela.saleem50@yahoo.com

Abstract

Linguistic typology is an area of linguistics concerned with analysis of and comparison between natural languages of the world based on their certain linguistic features. For that purpose, historically, the area has relied on manual extraction of linguistic feature values from textural descriptions of languages. This makes it a laborious and time expensive task and is also bound by human brain capacity. In this study, we present a deep learning system for the task of automatic extraction of linguistic features from textual descriptions of natural languages. First, textual descriptions are manually annotated with special structures called semantic frames. Those annotations are learned by a recurrent neural network, which is then used to annotate un-annotated text. Finally, the annotations are converted to linguistic feature values using a separate rule based module. Word embeddings, learned from general purpose text, are used as a major source of knowledge by the recurrent neural network. We compare the proposed deep learning system to a previously reported machine learning based system for the same task, and the deep learning system wins in terms of F1 scores with a fair margin. Such a system is expected to be a useful contribution for the automatic curation of typological databases, which otherwise are manually developed.

1 Introduction and Background

Linguistic typology is an area of linguistics concerned with analysis of and comparison between natural languages of the world in terms of their structural and functional attributes. Among others, major objectives of the area are exploring the range of possibilities for expressing different linguistic categories, trying to understand the extent to which the presence of different features depend on one another in larger patterns, and to study how the global distribution of language traits has come about through an interplay of tendencies inherent to languages and historical contingencies (Bickel, 2015). For achieving such aims, historically, the area has relied heavily on scholars having to read textual documents (commonly known as descriptive grammars) describing languages, manually extracting values of a pre-defined set of features, and then comparing languages based on the extracted feature values. To make the whole exercise systematic and structured, traditionally, the features are expressed in the form of questions e.g. 'What is the order of adjectives and noun in language X?', and scholar's job is then to find answers to such questions by reading descriptive documents about language X. The answers are often formulated to be simple strings e.g. 'NA', 'AN', or 'Both' representing the fact that nouns precede adjectives, nouns follow adjectives, or 'both nouns may follow or precede adjectives' respectively in the case of above give question. For easy storage and retrieval, the questions and their answers are recorded in special kind of databases known as typological databases. There exist many such databases and a fuller list is available at languagegoldmine.com/. These databases are later used to compare and analyze languages to achieve the above mentioned objectives of the area. In addition, information in such databases has proven to be useful for a number of natural language processing (NLP) related tasks. A survey on the usefulness of typological information in NLP can be found in (O'Horan et al., 2016).

As can be imagined, the manual development of typological databases is an expensive enterprise both in terms of time and efforts, and also their qual-

Proceedings of Recent Advances in Natural Language Processing, pages 1480–1489
Sep 1–3, 2021.
https://doi.org/10.26615/978-954-452-072-4_166

ity and coverage is bound by human brain capacities. Extensive digitization efforts (e.g. (Michel et al., 2011)) and the advancement of computational methodologies, including NLP, offer many possibilities for easing the task of developing linguistic typological databases. A maximally automated approach would allow for the generation of such databases at a hitherto unparalleled scale, increasing both the number of features and languages that can be analyzed and compared taking the area to new heights. However, this requires developing methodologies and systems for automatic extraction of typological linguistic information from descriptive documents. This exactly is a major objective of the study reported in this paper.

Previously, a few approaches and systems have been reported for the automatic extraction of typological information, but a practical system still remains to be be developed. Pattern matching and machine learning based classification are the two main computational paradigms that have been exploited so far (see Section 3). Pattern matching has its own limitations as it is impossible to design patterns which can cover every possible scenario. Similarly, machine learning classification algorithms are heavily dependent on feature engineering, which have its own limitations and features are often expensive to compute. To address some of these issues, we report a deep learning based system in this study (Section 4). First, textual descriptions are manually annotated (Section 4.1) with special structures called semantic frames, which are based on the theory of frame-semantics (Section 2). Those annotations are learned by a recurrent neural network using word embeddings learned from general purpose text as the major source of knowledge (Section 4.2). The trained model is used to annotate the un-annotated data, and the annotations are then converted to linguistic feature values using a separate rule based module (Section 8). Our frame annotation system beats previously reported systems on a test set with a fair margin in terms of F1-scores.

2 Frame-Semantics, FrameNet, and Frame-Semantic Parsing

Frame semantics is a theory of meanings in natural languages (Fillmore, 1982). It stipulates that meanings of words can be best understood with reference to the situations they invoke in the minds of the speakers. The concrete manifestation of frame semantics is a computational lexical resource called a framenet. The first such resource was the English Berkeley FrameNet (BNF) (Baker et al., 1998) which has inspired work on framenets for many other languages. The “lexical entry” in a framenet is called a (semantic) frame, which is a script-like description of a prototypical event, object, relation, or scenario. A frame consists of triggers – the lexical units (words) which evoke a specific situation – and additional components called frame-elements that fills in various semantic slots of the frame. Below is an example sentence manually labeled with the COMMERCE_SELLING semantic frame and its frame-elements i.e. seller, buyer, and price. The frame is triggered by the word ’sold’.

Jimmy $_{seller}$ [sold]$_{COMMERCE\_SELLING}$ [a car]$_{goods}$ [to Lester]$_{buyer}$ in [2000 USD]$_{price}$.

The process of automatically performing the above type of annotation/analysis is called semantic parsing. The first such frame semantic parser was proposed by Gildea and Jurafsky (2002), which has been followed by a number of other systems/approaches (e.g. (Das et al., 2014; Roth and Lapata, 2016)).

3 Related Work

Previously, a few experimental techniques and associated systems have been reported for automatic extraction of typological information. In (Borin et al., 2018; Virk et al., 2017), the authors have reported on simple pattern matching and syntactic parsing based systems. The systems have modest accuracy and recall and are very restricted with respect to the number of features they can target.

To overcome some of those limitations, in (Virk et al., 2019) the authors exploited frame-semantics and machine learning approaches to build a system that can extract information about a few experimental features. The systems is based on the work reported in (Malm et al., 2018) where the authors proposed to use frame-semantics to represent the typological linguistic information. They also developed a domain specific framenet (LingFN) containing semantic frames for the linguistic domain representing various linguistic terms and phenomenas (e.g. verb, noun, agreement, inflection, etc.). The developed LingFN was used by Virk et al. (2019) to annotate textual descriptions of languages with linguistic domain frames (more details in Section 4.1) and develop an automatic typlogical information extraction system using machine learning.

In (Søren and Rama, 2019), the authors reported a two step strategy to detect the parts of the text that possibly contains value of a given feature, and then extracting the feature value from it. For the first step their approach relies on keyword spotting and pattern matching, while on machine learning classification for the second step.

In (Hammarström, 2020), the author reported a simple keyword based approach for extracting values of one typological feature about tones for many languages from thousands of documents with an overall accuracy of 89.1%.

4 Proposed System

Figure 1 shows the complete architecture of the system that we propose. As shown, it has a clear division between three components: data annotation, deep learning, and typological information formulation. In the following subsection, the first two components will be explained, while the explanation of the third component is deliberately deferred until Section 8 for a better flow.

4.1 Data

A small corpus consisting of descriptive grammars of the natural languages spoken in South Asia was reported in (Borin et al., 2018), and a set of documents from that corpus annotated with LingFN frames was reported in (Virk et al., 2019). Annotation of a descriptive grammars with LingFN frames involve identification of lexical units and selection of appropriate linguistic semantic frames and their frame elements. As a part of the study reported in this paper, the annotated corpus was extended resulting in a total of 70 annotated documents (a document corresponds to 3 to 7 pages of text). Figure 2 shows an annotated sentence. As can be seen, the sentence is annotated with two linguistic domain frames i.e. VERB triggered by the word 'verb' having 'data' and 'data_translation' frame elements and the frame 'AFFIXATION' triggered by the word 'suffixed' and having 'degree' and 'anthromorphic_entity' frame elements. We refer the reader to (Virk et al., 2019) for more details on annotations and the annotation process. The data used in this study consists of a total of 70 documents comprising around 3,986 sentences, 7,170 semantic frames, and 4,669 frame-elements.

4.2 Deep Learning Part

In the system developed here, manually labeled data (reported in previous subsection) is used to train a deep learning model which is then used to label un-annotated data. The architecture of model is similar to the one proposed by Swayamdipta et al. (2017) using the RNNs. Figure 3 shows a simplified version of the architecture together with various inputs (features). To make the description self explanatory, we briefly explain here both the inputs and the architecture with respect to an example input sentence: 'Nouns agree with the adjectives'. The word 'agree' triggers the frame AGREEMENT (shown in red), while 'Nouns' and 'the adjectives' represent two text segments (in purple). These text segments fill in the roles of two frame-elements, i.e. 'Segment_1' and 'Segment_2', to be learned and later predicted by the deep learning model. The model uses two bilingual LSTM networks (biLSTM) and one LSTM for various computations as described below.

- Each word at position q in the input sentence is converted to a vector:

$$v_q = [d_q; e_q; o_q; \gamma_q] \quad (1)$$

  where d_q is the learned embedding[1] of the word type, e_q is a pre-trained embedding of the word type, o_q is the learned embedding of the part-of-speech tag of the word, and γ_q is the distance of the word from the beginning of the target (the word triggering a frame).

- These word representations are given as input to a bidirectional LSTM (biLSTM), each of whose hidden state then becomes a contextualized representation of the following form.

$$h_q^{tok} = [biLSTM^{tok}(v_1, v_2.....v_n)] \quad (2)$$

- These token representations are then used to compute contextualized representation of various spans of the sentence as shown below:

$$h_{(i,j)}^{span} = [biLSTM^{span}(h_1^{tok}, v_2^{tok}.....h_j^{tok})] \quad (3)$$

[1] In this study, we use Stanford's GloVe (Global Vectors for Word Representation) word embeddings (Pennington et al., 2014) as proposed by Swayamdipta et al. (2017). GloVe embeddings were created from data from Wikipedia and newswires.

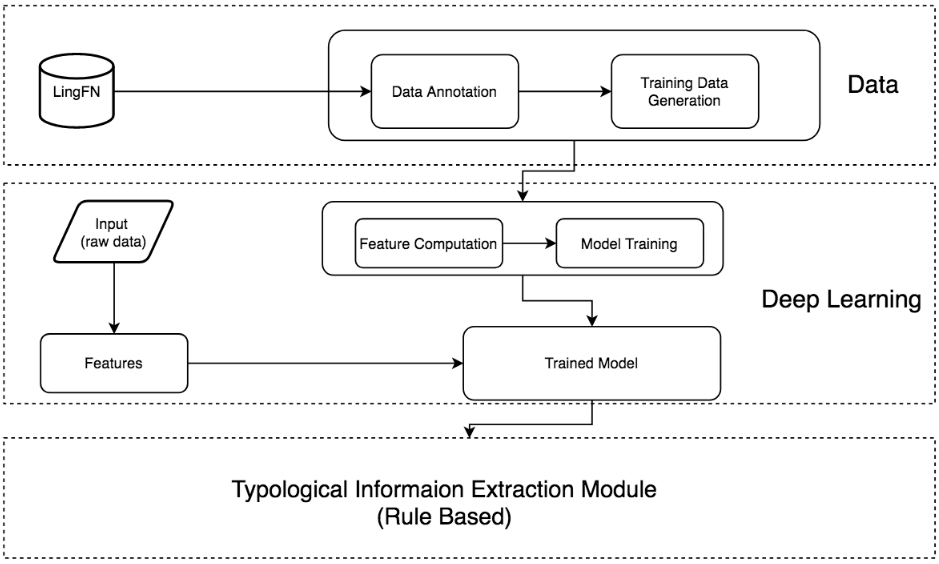
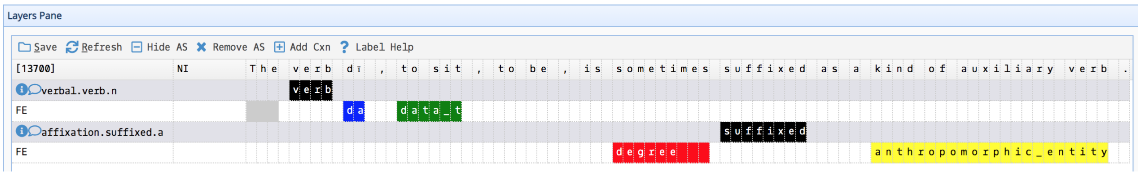


Figure 1: Complete architecture of the typological information extraction system



[13700]	NI	The verb dɪ , to sit , to be , is sometimes suffixed as a kind of auxiliary verb .
verbal.verb.n		verb
FE		da data_t
affixation.suffixed.a		suffixed
FE		degree anthropomorphic_entity

Figure 2: Screenshort of the web annotation tools with an example annotated sentence

- The token hidden states given above (Eq 2) together with an LSTM are used to compute a contextualized target representation as shown below:

$$v_t = [LSTM^{target}(h^{tok}_{t_{start}-1}, ...h^{tok}_{t_{start}+1})] \quad (4)$$

- The target representation, together with the learned frame embedding f_v and the lexical unit embedding v_l, are used to represent segments as:

$$v_{f,l,t} = [v_f; v_l; v_t] \quad (5)$$

- And, for every segment, a segment score is computed as:

$$v_s = [h^{span}_{i:j}; v_y; \mu] \quad (6)$$

Where v_y is a learned embedding of a segment at position (i, j) and μ represents two other features: the length of the span, and the span's position with respect to the target.

This is then passed through a rectified linear unit to get a segment score as:

$$\phi(s,x) = w2.reLU\{w1[v_s; v_{f,l,t}]\} \quad (7)$$

The segment score then becomes part of a criterion, which the model is trained to maximize on the training data. Once trained, the model is used to predict labels for various spans of the input sentence. The experiments to label sentences using the learned models are reported in the next section. For more technical details on the deep learning model and its input, we refer the reader to (Swayamdipta et al., 2017).

5 Experiments

The data was divided into two major sets labeled as 'Full' and 'Filtered'. The former is the data set where the annotation of all frames and frame-elements were preserved, while in the later the annotation of two problematic frame elements (i.e. 'data' and 'data_translation'[2]) were removed from the training data. The reason for removing

[2] In LingFN, many frames have data and data_translation frame-elements, which are used to represent examples of various morphology or grammatical linguistic categories of the language described in the document. As an example consider the following annotation in which both data and data_translation frame-elements have been used to annotate an example numeral (i.e. ghrī) and its translation (i.e. 'one'): *'The numeral [ghrī]_data , [one]_data_translation , is used as an indefinite article.*

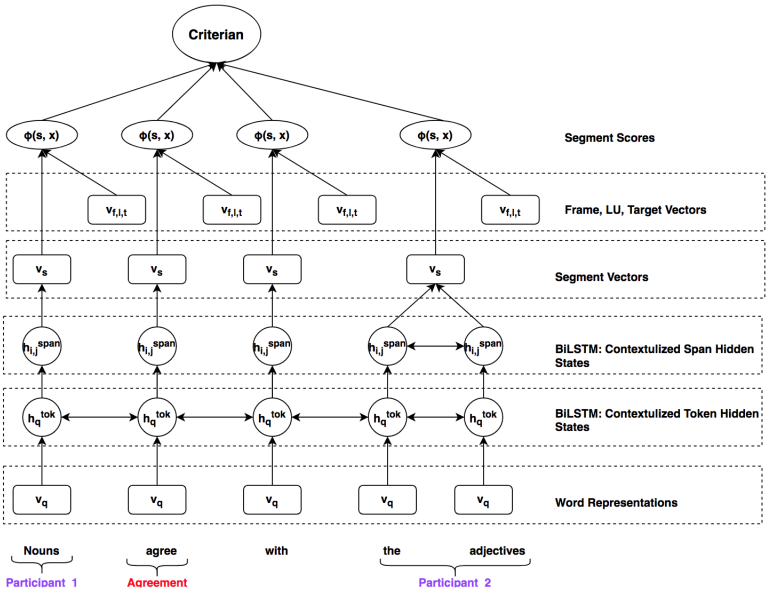


Figure 3: Deep Learning Model Architecture and Inputs

these two frame-elements is that they contain non-English words for which the embeddings are missing in the learned embedding space consequently impacting the model's performance adversely. To show their impact, we conducted experiments with and without those frame-element annotations. With those two data sets, the experiments were conducted with the following three settings:

- Word Embeddings
- Character Embeddings
- Mimic Embeddings

In the first setting, word-embeddings were used to compute the token level representations (i.e. Eq 1), and consequently all other computations involving token representations. In the second setting, word-embeddings were replaced with character embedding, while in the third setting they were replaced with mimic embeddings. The motivation behind these three settings follows.

Since the word embedding space was learned from a general-purpose text, it understandably does not contain many of the the domain specific words in the data used in this study (i.e. the descriptive grammars). In addition, there are the non-English words, i.e the transliterated forms of the words of a specific language given as examples in the descriptive documents. In the annotated data, such words often appear as a part of the 'data' frame-element. For all those cases, a default word embedding was used by the system while learning and predicting the frame annotations. This means that the knowledge that words can bring to the system while learning was not available to the system resulting in a decreased system performance (as obvious from the results discussed in Section 6). As a solution to this out-of-vocabulary words issue, we experimented with the other two settings.

In the second setting, word embeddings were replaced with character embeddings while computing Eq 1, resulting in different token and span level representations (Eq 2 and 3). This was based on the intuition that even if a complete word has not been seen in the training set, the characters it contains have, hence, utilizing the character-level knowledge. This is particularly pertinent in the case of 'data' frame element. As is apparent from the results in given in Section 6, this technique improved the result considerably. A primary computational concern with this solution from the outset was the need to train character embeddings, which is computationally a highly expensive task. For that we relied on automatic conversion of a set of pre-trained word embeddings into character embeddings[3]. In this technique, for each word in a set of word embeddings, each character is given a vector inferred from the parent word. Then, for each word in the embeddings, when the character is seen again this vector is adjusted to reflect its average in the entire set. This technique is a useful workaround to get a meaningful set of character embeddings.

[3] https://minimaxir.com/2017/04/char-embeddings/

In the third setting, word embeddings of the unknown words were inferred from the existing word-embedding space and used while computing input (Eq 1). The embedding inferring technique has been proposed by Pinter et al. (2017) in which a character-based bidirectional LSTM was used to infer vectors for a list of unknown words and to add them to the existing word-embeddings. They call their method the 'mimick', and it predicts an embedding for a word that should fit into the same space based on the sequence of character embeddings for the characters in the given word.

Originally, the authors used their technique in the context of OCR error correction and spelling variations issue, but in our case we have used it for the out-of-vocabulary issue. It appeared to be equally useful in our case as is obvious from the results given in the next section.

6 Results

Table 1 shows the results of three experimental settings discussed in the previous section. To show the impact of the 'data' and 'data_translation' frame-elements, results have been shown with and without them (i.e. 'Full' and 'Filtered' in the table). As can be seen, the system achieved a considerably low F-score when tested on the full data, which improved when the data and data_translation frame-elements were removed. The word to character level embedding replacement obtained an improvement from 46.9 to 51.3 on the full data, while from 55.0 to 58.8 without data and data_translation frames. This suggests that the character-level embeddings are a better choice in a domain specific setting even if there are no non-English transliterated words. As for the mimic technique, it deteriorates from 46.9 to 46.3 when applied on the full data, but improves from 55.0 to 57.0 when tested without the problematic data and data_translation frame elements. This suggests:

- The technique of inferring word embeddings and using them for the out-of-vocabulary issue is equally useful as it was for the OCR error and spelling variation issue.

- It is not advantageous to infer and use the embeddings of totally unrelated words i.e. the non-English transliterated words, such as words in the data frame element.

In summary, we achieved the best results with character-level embedding when applied to the full data, and with the mimic technique when data and data_translation were excluded.

7 Comparison to a Previous System

Table 2 shows comparison between the deep learning based vs a machine learning based system reported in (Virk et al., 2019) for the frame annotation task. The comparison is done on a separate smaller data set as the (Virk et al., 2019) system could not be run on the full data set due to memory issues arising from data size. As can be seen, both on the full and filtered data set versions, the proposed system beats the older system with a fair margin in term of F-scores. This proves the worth of such a system for domain specific frame semantic parsing which is to be used for typological information extraction as described in the next section.

8 Typological Information Formulation Module

In the proposed system, once a sentence has been automatically annotated with semantic frames and their frame-elements (the output from the second part of the system architecture), the annotations can be converted to a typolological feature value as an answer to a typological question. Currently, we rely on a rule-based module for the conversion from annotations to feature values as explained in (Virk et al., 2019). This involves writing small modules as shown in Algorithm 1 given in Appendix A for converting annotations to feature values.

As can be seen the algorithm simply loops through the set of frames (line 2) and frame elements, and depending on their contents (line 3, 7, 8, 10, and 12), it assigns appropriate features value for the adjective-noun-order feature (line 9, 11, and 13) discussed in the introduction section. Similar modules can be used for other typological features.

Appendix B shows a set of order related feature and their values automatically extracted from descriptive grammar of 'Ulwa' language (Barlow, 2018) using the developed semantic parser, and the rule-based feature formulation module given in Appendix A. Note all features were extracted using only SEQUENCE[4] semantic frame from LingFN.

[4] A semantic frame in LingFN which encodes ordering related information similar to the example given in the introduction section.

Setting	Data	Precision	Recall	F-score
Word-Embeddings (baseline)	Full	0.52459	0.42440	0.46921
Word-Embeddings (baseline)	Filtered	0.60000	0.50847	0.55046
Character-Embeddings	Full	0.62595	0.43501	0.51330
Character-Embeddings	Filtered	0.65341	0.48729	0.55825
Mimic	Full	0.58233	0.38462	0.46326
Mimic	Filtered	0.61165	0.53390	**0.57014**

Table 1: Experimental Results

System	Data	F-score
Virk et el	Full	35.6
Character-Embedd	Filtered	45.9
Virk et el	Full	52.9
Character-Embedd	Filtered	**62.9**

Table 2: Comparison to a previously reported system

9 Conclusions and Future Work

Our main contributions are two-fold. First, we have reported a deep learning based system for the automatic extraction of typological information from descriptive grammars of natural languages. As mentioned previously, the manual extraction of such information is very costly both in terms of cost and human efforts. Any assistance in this regard is much appreciated as typological linguistic information is not only useful for investigating the linguistic diversity of the universe, but is also being used for many other NLP related tasks. A survey of usefulness of typological information in various NLP tasks can be found in (O'Horan et al., 2016). Unlike, previously reported systems for the same task, the system proposed in this study uses word-embeddings as the only knowledge source and does not require any feature engineering to identify suitable feature set for the machine learning part.

Second, we have shown how word-embeddings learned from general purpose text can be used in a domain specific setting, and how character embeddings can be be used as a work around for out-of-vocabulary terms. Further, inferring word embeddings is another way to deal with out of vocabulary words as long as the words are not cross-lingual (English and non-English in our case). In the future, we would like to improve the system by experimenting with n-gram embedding instead of character embeddings. In another direction, word-embeddings could be learned from domain-specific data-sets as opposed to general purpose word-embeddings used in this study, which could avoid the issue of out-of-vocabulary words issue.

Evaluation of the extracted typological information is another task that we have plans to carry out in the near future. One can select a suitable set of features from one of the existing typological databases, and use their values as a gold-standard to evaluate the performance of the system proposed in this study.

Acknowledgements

The work presented here was funded by (1) the *Dictionary/Grammar Reading Machine: Computational Tools for Accessing the World's Linguistic Heritage* (DReaM) Project awarded 2018–2020 by the Joint Programming Initiative in Cultural Heritage and Global Change, Digital Heritage and Riksantikvarieämbetet, Sweden; (2) the Swedish Research Council as a part of the project *South Asia as a linguistic area? Exploring big-data methods in areal and genetic linguistics* (2015–2019, dnr 421-2014-969); (3) *From Dust to Dawn: Multilingual Grammar Extraction from Grammars* project funded by Stiftelsen Marcus och Amalia Wallenbergs Minnesfond 2007.0105, Uppsala University; (4) the Swedish Research Council as part of the project *Evaluation and refinement of an enhanced OCR-process for mass digitisation* (2019–2020, dnr IN18-0940:1 and 421-2014-969). It is also supported by Språkbanken Text and Swe-Clarin, a Swedish consortium in Common Language Resources and Technology Infrastructure (CLARIN) Swedish CLARIN (dnr 821-2013-2003).

References

Collin F. Baker, Charles J. Fillmore, and John B. Lowe. 1998. The Berkeley FrameNet project. In *Proceedings of ACL/COLING 1998*. ACL, Montreal, pages 86–90. https://doi.org/10.3115/980845.980860.

Russell Barlow. 2018. *A grammar of Ulwa*. Ph.D. thesis, University of Hawai'i at Mānoa.

Balthasar Bickel. 2015. Distributional typology: statistical inquiries into the dynamics of linguistic diversity. In Bernd Heine and Heiko Narrog, editors, *The Oxford Handbook of Linguistic Analysis*, Oxford: Oxford University Press, pages 901–923. 2 edition.

Lars Borin, Shafqat Mumtaz Virk, and Anju Saxena. 2018. Language technology for digital linguistics: Turning the Linguistic Survey of India into a rich source of linguistic information. In Alexander Gelbukh, editor, *Computational Linguistics and Intelligent Text Processing*. Springer, Cham, pages 550–563.

Dipanjan Das, Desai Chen, André F. T. Martins, Nathan Schneider, and Noah A. Smith. 2014. Frame-semantic parsing. *Computational Linguistics* 40:1:9–56.

Charles J. Fillmore. 1982. Frame semantics. In Linguistic Society of Korea, editor, *Linguistics in the Morning Calm*, Hanshin Publishing Co., Seoul, pages 111–137.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computational Linguistics* 28(3):245–288. https://doi.org/10.1162/089120102760275983.

Harald Hammarström. 2020. Keyword spotting: A quick-and-dirty method for extracting typological features of language from grammatical descriptions. In *SIGTYP Workshop: THE ACL SPECIAL INTEREST GROUP ON TYPOLOGY (SIGTYP)*.

Per Malm, Malin Ahlberg, and Dan Rosén. 2018. Uneek: A web tool for comparative analysis of annotated texts. In *Proceedings of the IFNW 2018 Workshop on Multilingual FrameNets and Constructicons at LREC 2018*. ELRA, Miyazaki, pages 33–36.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva Presser Aiden, Adrian Veres, Matthew K. Gray, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden. 2011. Quantitative analysis of culture using millions of digitized books. *Science* 331(6014):176–182.

Helen O'Horan, Yevgeni Berzak, Ivan Vulić, Roi Reichart, and Anna Korhonen. 2016. Survey on the use of typological information in natural language processing. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*. The COLING 2016 Organizing Committee, Osaka, Japan, pages 1297–1308. https://www.aclweb.org/anthology/C16-1123.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Association for Computational Linguistics, Doha, Qatar, pages 1532–1543. https://doi.org/10.3115/v1/D14-1162.

Yuval Pinter, Robert Guthrie, and Jacob Eisenstein. 2017. Mimicking word embeddings using subword RNNs. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Copenhagen, Denmark, pages 102–112. https://doi.org/10.18653/v1/D17-1010.

Michael Roth and Mirella Lapata. 2016. Neural semantic role labeling with dependency path embeddings. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Berlin, Germany, pages 1192–1202. https://doi.org/10.18653/v1/P16-1113.

Swabha Swayamdipta, Sam Thomson, Chris Dyer, and Noah A. Smith. 2017. Frame-semantic parsing with softmax-margin segmental RNNs and a syntactic scaffold. *CoRR* abs/1706.09528.

Wichmann Søren and Taraka Rama. 2019. Towards unsupervised extraction of linguistic typological features from language descriptions. In *First Workshop on Typology for Polyglot NLP, Florence,*.

Shafqat Virk, Lars Borin, Anju Saxena, and Harald Hammarström. 2017. Automatic extraction of typological linguistic features from descriptive grammars. In *Proceedings of TSD 2017*. Springer, Cham, pages 111–119.

Shafqat Mumtaz Virk, Azam Sheikh Muhammad, Lars Borin, Muhammad Irfan Aslam, Saania Iqbal, and Nazia Khurram. 2019. Exploiting frame-semantics and frame-semantic parsing for automatic extraction of typological information from descriptive grammars of natural languages. In *RANLP-Proceedings*. pages 1247–1256.

A Algorithm 1

```
procedure ExtractAdjectiveNounOrder($parse$)
    for <every frame in parse> do
        if $frame = SEQUENCE$ then
            $NA \leftarrow False$
            $AN \leftarrow False$
            $Both \leftarrow False$
            if $'adjective' \in Entity\_1 \wedge' noun' \in Entity\_2$ then
                if $Frequency \in [sometimes, usually, mostly, often]$ then
                    $Both \leftarrow True$
                else if $order = follow$ then
                    $AN \leftarrow True$
                else if $order = precede$ then
                    $NA \leftarrow True$
                end if
            end if
        end if
    end for
end procedure
```

B Extracted Typological Information

Feature	Value
Subject and NP Order	NP–SubjectMarker
Object and NP Order	NP–ObjectMarker
Constituent Order	SOV
PostpositionalPhrase–Oblique-markedNP Order	Both
ObliguePhrase–SubjectOFClause Order	SubjectOFClause-ObliguePhrase
ObliguePhrase–Verb Order	ObliguePhrase–Verb
Negator–Verb Order	Negator–Verb
AdPosition–NP Order	NP–AdPosition
Possessor–Possessum Order	Possessor–Possessum
Adjective–Noun Order	Noun–Adjective
Demonstrative–Noun Order	Noun–Demonstrative
Numeral–Noun Order	Noun–Numeral
RelativeClause–HeadNoun Order	RelativeClause–HeadNoun
PossessivePronoun–Noun Order	PossessivePronoun–Noun
ObliqueMarker–Noun Order	Noun–ObliqueMarker
TransitiveVerb–ObjectMarker Order	TransitiveVerb–ObjectMarker
NominalizedVerb–SubjectMarker Order	SubjectMarker–NominalizedVerb
Verb–DirectObject Order	DirectObject–Verb
Oblique–Verb Order	Oblique–Verb
Oblique- Subject Order	Subject–Oblique
Adverb–Subject Order	Subject–Adverb
Adverb–Object Order	Adverb–Object
Adverb–Oblique-markedNP Order	Adverb–Oblique-markedNPs
NasalSegments–VoicelessStops Order	NasalSegments–VoicelessStops
LabialNasal–PalatoAlveolar Order	LabialNasal–PalatoAlveolar
HomorganicNasals–VoicelessStops Order	HomorganicNasals–VoicelessStops
Liquids–LabialStops Order	LabialStops–Liquids