Neural Surface Realization for Italian

Valerio Basile
Dipartimento di Informatica
Università degli Studi di Torino
Corso Svizzera 185, 10153 Torino
basile@di.unito.it.com

Alessandro Mazzei
Dipartimento di Informatica
Università degli Studi di Torino
Corso Svizzera 185, 10153 Torino
mazzei@di.unito.it

Abstract

We present an architecture based on neural networks to generate natural language from unordered dependency trees. The task is split into the two subproblems of word *order prediction* and *morphology inflection*. We test our model gold corpus (the Italian portion of the Universal Dependency treebanks) and an automatically parsed corpus from the Web.

(Italian) Questo lavoro introduce un'architettura basata su reti neurali per generare frasi in linguaggio naturale a partire da alberi a dipendenze. Il processo è diviso nei due sotto-problemi dell'ordinamento di parole e dell'inflessione morfologica, per i quali la nostra architettura prevede due modelli indipendenti, il cui risultato è combinato nella fase finale. Abbiamo testato il modello usando un gold corpus e un silver corpus ottenuto dal Web.

1 Introduction

Natural Language Generation is the process of producing natural language utterances from an abstract representation of knowledge. As opposed to Natural Language Understanding, where the input is well-defined (typically a text or speech segment) and the output may vary in terms of complexity and scope of the analysis, in the generation process the input can take different forms and levels of abstraction, depending on the specific goals and applicative scenarios. However, the input structures for generation should be at least formally defined.

In this work we focus on the final part of the standard NLG pipeline defined by Reiter and Dale (2000), that is, *surface realization*, the task of producing natural language from formal abstract representations of sentences' meaning and syntax.

We consider the surface realization of unordered Universal Dependency (UD) trees, i.e., syntactic structures where the words of a sentence are connected by labeled directed arcs in a tree-like fashion. The labels on the arcs indicate the syntactic relation holding between each word and its dependent words (Figure 1a). We approach the surface realization task in a supervised statistical setting. In particular, we draw inspiration from Basile (2015) by dividing the task into the two independent subtasks of **word order** prediction and **morphology inflection** prediction. Two neural network-based models run in parallel on the same input structure, and their output is later combined to produce the final surface form.

A first version of the system implementing our proposed architecture (called the *DipInfo-UniTo realizer*) was submitted to the shallow track of the *Surface Realization Shared Task 2018* (Mille et al., 2018). The main research goal of this paper is to provide a critical analysis for tuning the training data and learning parameters of the DipInfo-UniTo realizer.

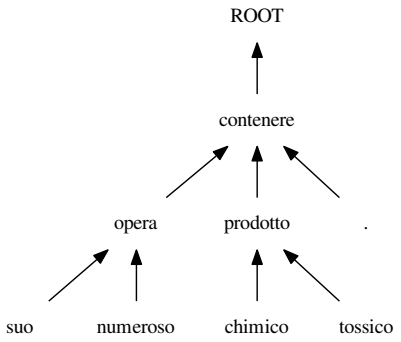
2 Neural network-based Surface Realization

In the following sections, we detail the two neural networks employed to solve the subtasks of word order prediction (2.1) and morphology inflection (2.2) respectively.

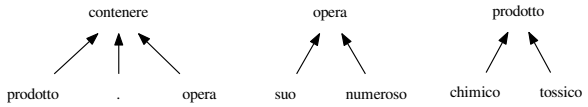
2.1 Word Ordering

We reformulate the problem of sentence-wise word ordering in terms of reordering the subtrees of its syntactical structure. The algorithm is composed of three steps: i) splitting the unordered tree into single-level unordered subtrees; ii) predicting the local word order for each subtree; iii) recomposing the single-level ordered subtrees into a single multi-level ordered tree to obtain the global word order.

In the first step, we split the original unordered universal dependency multilevel tree into a number of single-level unordered trees, where each subtree is composed by a head (the root) and all its dependents (the children), similarly to Bohnet et al. (2012). An example is shown in Figure 1:



(a) Tree corresponding to the Italian sentence "Numerose sue opere contengono prodotti chimici tossici." (*"Many of his works contain toxic chemicals."*)



(b) Three subtrees extracted from the main tree.

Figure 1: Splitting the input tree into subtrees to extract lists of items for learning to rank.

from the (unordered) tree representing the sentence "*Numerose sue opere contengono prodotti chimici tossici.*" (1a), each of its component subtrees (limited to one-level dependency) is considered separarately (1b). The head and the dependents of each subtree form an unordered list of lexical items. Crucially, we leverage the flat structure of the subtrees in order to extract structures that are suitable as input to the learning to rank algorithm in the next step of the process.

In the second step of the algorithm, we predict the relative order of the head and the dependents of each subtree with a *learning to rank* approach. We employ the list-wise learning to rank algorithm *ListNet*, proposed by Cao et al. (2007). The relatively small size of the lists of items to rank allows us to use a list-wise approach, as opposed to pair-wise or poin-twise approaches, while keeping the computation times manageable. ListNet uses a list-wise loss function based on *top one probability*, i.e., the probability of an element of being the first one in the ranking. The top one probability model approximates the *permutation probability* model that assigns a probability to each possible permutation of an ordered list. This approximation is necessary to keep the problem tractable by avoiding the exponential explosion of the number of permutations. Formally, the top one probability of an object j is defined as

$$P_s(j) = \sum_{\pi(1)=j, \pi \in \Omega_n} P_s(\pi)$$

that is, the sum of the probabilities of all the possible permutations of n objects (denoted as Ω_n) where j is the first element. $s = (s_1, ..., s_n)$ is a given list of *scores*, i.e., the position of elements in the list. Considering two permutations of the same list y and z (for instance, the predicted order and the reference order) their distance is computed using cross entropy. The distance measure and the top one probabilities of the list elements are used in the loss function:

$$L(y, z) = -\sum_{j=1}^{n} P_y(j) log(P_z(j))$$

The list-wise loss function is plugged into a linear neural network model to provide a learning environment. ListNet takes as input a sequence of ordered lists of feature vectors (the features are encoded as numeric vectors). The weights of the network are iteratively adjusted by computing a list-wise cost function that measure the distance between the reference ranking and the prediction of the model and passing its value to the gradient descent algorithm for optimization of the parameters.

The choice of features for the supervised learning to rank component is a critical point of our solution. We use several word-level features encoded as one-hot vectors, namely: the universal POS-tag, the treebank specific POS tag, the morphology features and the head-status of the word (head of the single-level tree vs. leaf). Furthermore, we included word representations, differentiating between content words and function words: for open-class word lemmas (content words) we added the corresponding language-specific word embedding to the feature vector, from the pretrained multilingual model Polyglot (Al-Rfou' et al., 2013). Closed-class word lemmas (function words) are encoded as one-hot bags of words vectors. An implementation of the feature encoding for the word ordering module of our architecture is available online[1].

[1] https://github.com/alexmazzei/ud2ln

In the third step of the word ordering algorithm, we reconstruct the global (i.e. sentence-level) order from the local order of the one-level trees under the hypothesis of projectivity[2] — see Basile and Mazzei (2018) for details on this step.

2.2 Morphology Inflection

The second component of our architecture is responsible for the morphology inflection. The task is formulated as an alignment problem between characters that can be modeled with the *sequence to sequence* paradigm. We use a deep neural network architecture based on a hard attention mechanism. The model has been recently introduced by Aharoni and Goldberg (2017). The model consists of a neural network in an encoder-decoder setting. However, at each step of the training, the model can either write a symbol to the output sequence, or move the attention pointer to the next state of the sequence. This mechanism is meant to model the natural monotonic alignment between the input and output sequences, while allowing the freedom to condition the output on the entire input sequence.

We employ all the morphological features provided by the UD annotation and the dependency relation binding the word to its head, that is, we transform the training files into a set of structures $((lemma, features), form)$ in order to learn the neural inflectional model associating a $(lemma, features)$ to the corresponding $form$. An example of training instance for our morphology inflection module is the following:

```
lemma: artificiale
features:
  uPoS=ADJ
  xPoS=A
  rel=amod
  Number=Plur
form: artificiali
```

Corresponding to the word form *artificiali*, an inflected form (plural) of the lemma *artificiale* (artificial).

3 Evaluation

In this section, we present an evaluation of the models presented in Section 2, with particular consideration for two crucial points influencing the performances of the DipInfo-UniTo realizer, namely training data and learning parameters settings. In Basile and Mazzei (2018), the hardware limitations did not allow for an extensive experimentation dedicated to the optimization of the realizer performances. In this paper, we aim to bridge this gap by experimenting with higher computing capabilities, specifically a virtualized GNU/Linux box with 16-core and 64GB of RAM.

3.1 Training Data

For our experiments, we used the four Italian corpora annotated with Universal Dependencies available on the Universal Dependency repositories[3]. In total, they comprise 270,703 tokens and 12,838 sentences. We have previously used this corpus for the training of the DipInfo-UniTo realizer that participated to the SRST18 competition (Basile and Mazzei, 2018). We refer to this corpus as *Gold-SRST18* henceforth.

Moreover, we used a larger corpus extracted from ItWaC, a large unannotated corpus of Italian (Baroni et al., 2009). We parsed ItWaC with UDpipe (Straka and Straková, 2017), and selected a random sample of 9,427 sentence (274,115 tokens). We refer to this corpus as *Silver-WaC* henceforth.

3.2 Word Ordering Performances

We trained the word order prediction module of our system[4] on the Gold-SRST18 corpus as well as on the larger corpus created by concatenating Gold-SRST18 and Silver-WaC.

The performance of the ListNet algorithm for word ordering is given in terms of average Kendall's Tau (Kendall, 1938, τ), a measure of rank correlation used to give a score to each of the rankings predicted by our model for every subtree (Figure 2). τ measures the similarity between two rankings by counting how many pairs of elements are swapped with respect to the original ordering out of all possible pairs of n elements:

$$\tau = \frac{\#concordant\_pairs - \#discordant\_tpairs}{\frac{1}{2}n(n-1)}$$

Therefore, τ ranges from -1 to 1.

In Figure 2 we reported the τ values obtained at various epochs of learning for both the Gold-

[2] As a consequence of the design of our approach, the DipInfo-UniTo realizer cannot predict the correct word order for non-projective sentences.

[3] http://universaldependencies.org/

[4] Our implementation of ListNet featuring a regularization parameter to prevent overfitting is available at https://github.com/valeriobasile/listnet

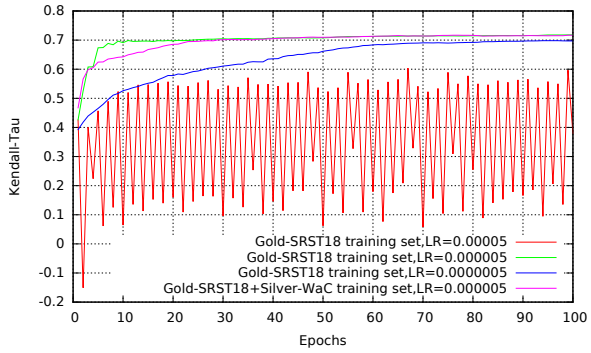


Figure 2: The trend of the τ value with respect to the ListNet iteration.

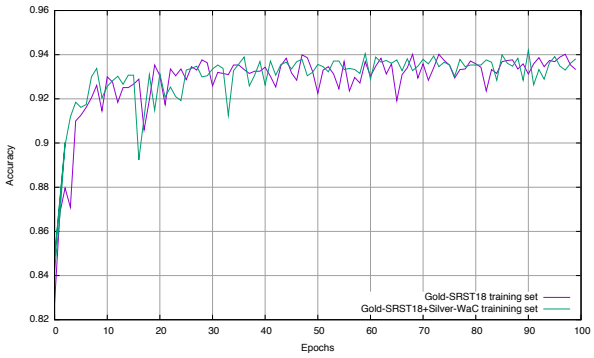


Figure 3: The trend of the Morphology Accuracy on the SRST18 development set with respect to the DNN training epochs.

SRST18 and Gold-SRST18+Silver-WaC corpora. In particular, in order to investigate the influence of the learning rate parameter (LR) in the learning of the ListNet model, we reported the τ trends for $LR = 5 \cdot 10^{-5}$ (the value originally used for the official SRST18 submission), $LR = 5 \cdot 10^{-6}$ and $LR = 5 \cdot 10^{-7}$. It is quite clear that the value of LR has a great impact on the performance of the word ordering, and that $LR = 5 \cdot 10^{-5}$ is not appropriate to reach the best performance. This explains the poor performance of the DipInfo-UniTo realizer in the SRST18 competition (Table 1). Indeed, the typical zigzag shape of the curve suggests a sort of loop in the gradient learning algorithm. In contrast, the $LR = 5 \cdot 10^{-6}$ seems to reach a plateau value after the 100th epoch with both corpora used in the experiments. We used the system tuned with this value of the learning rate to evaluate the global performance of the realizer.

3.3 Morphology Inflection Performances

In order to understand the impact of the Silver-WaC corpus on the global performance of the system, we trained the DNN system for morphology inflection[5] both on the Gold-SRST18 corpus and on the larger corpus composed by Gold-SRST18+ Silver-WaC. In Figure 3 we reported the accuracy on the SRST18 development set for both the corpora. A first analysis of the trend shows little improvement to the global performance of the realization from the inclusion of additional data (see the discussion in the next section).

3.4 Global Surface Realization Performances

Finally, we evaluate the end-to-end performance of our systems by combining the output of the two modules and submitting it to the evaluation scorer of the Surface Realization Shared Task. In Table 1 we report the performance of various tests systems with respect to the BLUE-4, DIST, NIST measures, as defined by Mille et al. (2018). The first line reports the official performance of the DipInfo-Unito realizer in the SRST18 for Italian. The last line reports the best performances achieved on Italian by the participants to SRST18 (Mille et al., 2018). The other lines report the performance of the DipInfo-UniTo realizer by considering various combination of the gold and silver corpora. The results show a clear improvement

ListNet	**Morpho**	BLEU-4	DIST	NIST
G^{srst}	G^{srst}	24.61	36.11	8.25
G	G	36.40	32.80	9.27
G	G+S	36.60	32.70	9.30
G+S	G	36.40	32.80	9.27
G+S	G+S	36.60	32.70	9.30
-	-	44.16	58.61	9.11

Table 1: The performances of the systems with respect to the BLUE-4, DIST, NIST measures.

for the word order module (note that the DIST metric is character-based, therefore it is more sensitive to the morphological variation than NIST and BLEU-4). In contrast, the morphology submodule performance seems to be unaffected by the use of a larger training corpus. This effect could be due different causes. Errors are present in the silver standard training set, and it is not clear to what extent the morphology analysis is correct

[5] An implementation of the model by (Aharoni and Goldberg, 2017) is freely available as `https://github.com/roeeaharoni/morphological-reinflection`

with respect to the syntactic analysis. The other possible cause is the neural model itself. Indeed, Aharoni and Goldberg (2017) report a plateau in performance after feeding it with relatively small datasets. The DipInfo-UniTo realizer performs better than the best systems of the SRST18 challenge for one out of three metrics (NIST).

4 Conclusion and Future Work

In this paper, we considered the problem of analysing the impact of the training data and parameters tuning on the (modular and global) performance of the DipInfo-UniTo realizer. We computationally proved that the DipInfo-UniTo realizer can gives competitive results (i) by augmenting the training data set with automatically annotated sentences, and (ii) by tuning the learning parameters of the neural models.

In future work, we intend to resolve the main lack of our approach, that is the impossibility to realize non-projective sentences. Moreover, further optimization of both neural models will be carried out on a new high-performance architecture (Aldinucci et al., 2018), by executing a systematic grid-search over the hyperparameter space, namely the regularization factor and weight initialization for ListNet, and the specific DNN hyperparameters for the morphology module.

Aknowledgment

We thank the GARR consortium which kindly allowed to use to the GARR Cloud Platform[6] to run some of the experiments described in this paper. Valerio Basile was partially funded by Progetto di Ateneo/CSP 2016 (*Immigrants, Hate and Prejudice in Social Media*, S1618_L2_BOSC_01).
Alessandro Mazzei was partially supported by the HPC4AI project, funded by the Region Piedmont POR-FESR 2014-20 programme (INFRA-P call).

References

Roee Aharoni and Yoav Goldberg. 2017. Morphological inflection generation with hard monotonic attention. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017*, pages 2004–2015.

Rami Al-Rfou', Bryan Perozzi, and Steven Skiena. 2013. Polyglot: Distributed word representations for multilingual nlp. In *CoNLL*, pages 183–192. ACL.

[6] https://cloud.garr.it

Marco Aldinucci, Sergio Rabellino, Marco Pironti, Filippo Spiga, Paolo Viviani, Maurizio Drocco, Marco Guerzoni, Guido Boella, Marco Mellia, Paolo Margara, Idillio Drago, Roberto Marturano, Guido Marchetto, Elio Piccolo, Stefano Bagnasco, Stefano Lusso, Sara Vallero, Giuseppe Attardi, Alex Barchiesi, Alberto Colla, and Fulvio Galeazzi. 2018. Hpc4ai, an ai-on-demand federated platform endeavour. In *ACM Computing Frontiers*, Ischia, Italy, May.

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. 2009. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226, September.

Valerio Basile and Alessandro Mazzei. 2018. The dipinfo-unito system for srst 2018. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 65–71. Association for Computational Linguistics.

Valerio Basile. 2015. *From Logic to Language : Natural Language Generation from Logical Forms.* Ph.D. thesis, University of Groningen, Netherlands.

Bernd Bohnet, Anders Björkelund, Jonas Kuhn, Wolfgang Seeker, and Sina Zarrieß. 2012. Generating non-projective word order in statistical linearization. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 928–939. Association for Computational Linguistics.

Zhe Cao, Tao Qin, Tie-Yan Liu, Ming-Feng Tsai, and Hang Li. 2007. Learning to rank: From pairwise approach to listwise approach. In *Proceedings of the 24th International Conference on Machine Learning*, ICML '07, pages 129–136, New York, NY, USA. ACM.

M. G. Kendall. 1938. A new measure of rank correlation. *Biometrika*, 30(1/2):81–93.

Simon Mille, Anja Belz, Bernd Bohnet, Yvette Graham, Emily Pitler, and Leo Wanner. 2018. The first multilingual surface realisation shared task (sr'18): Overview and evaluation results. In *Proceedings of the First Workshop on Multilingual Surface Realisation*, pages 1–12. Association for Computational Linguistics.

Ehud Reiter and Robert Dale. 2000. *Building Natural Language Generation Systems*. Cambridge University Press, New York, NY, USA.

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, Vancouver, Canada, August. Association for Computational Linguistics.