Computational Linguistics and Intellectual Technologies:
Proceedings of the International Conference “Dialogue 2019”

Moscow, May 29—June 1, 2019

AN ATTENTION-BASED APPROACH TO AUTOMATIC GAPPING RESOLUTION FOR RUSSIAN

Movsesyan A. A. (derise@iitp.ru)
Institute for Information Transmission Problems (Kharkevich Institute), Russian Academy of Sciences, Moscow, Russia

Gapping is a type of ellipsis in which a finite verb is elided in a coordinate structure. Reconstruction of the elided material is essential for different NLP tasks. However, from a practical point of view, the problem did not receive considerable attention for Russian language because of lack of training data and rarity of the phenomenon itself. This paper is one of the first works of deep learning-based automatic gapping resolution in Russian as a part of AGRR-2019 competition. We used a recurrent neural network-based approach to determine presence/absence of gapping in a sentence and for the full annotation we applied a Universal Transformer neural network that combines self-attention mechanism with recurrence in depth. Also using pre-trained fastText word embeddings, we achieved 85% standard F-measure on test set for binary classification task and 62% symbol-wise F-measure for full annotation task. We assume that fixed word embedding like fastText does not contain enough syntactic information to properly match remnants in sentences with gapping. Also we show that our model generalize better if punctuation marks were ignored during training and evaluation.

Key words: gapping, Universal Transformer, fastText, deep learning, NLP

АВТОМАТИЧЕСКОЕ РАЗРЕШЕНИЕ ЯВЛЕНИЯ ГЭППИНГА ДЛЯ РУССКОГО ЯЗЫКА С ИСПОЛЬЗОВАНИЕМ МЕХАНИЗМА ВНИМАНИЯ

Мовсесян А. А. (derise@iitp.ru)
Институт проблем передачи информации РАН
им. А. А. Харкевича, Москва, Россия

Гэппингом называют тип эллипсиса, когда при сочинении опускается финитный глагол. Восстановление опущенного предиката является важным для различных задач обработки естественного языка. Однако, эта задача, с практической точки зрения, не привлекала существенного внимания исследователей ввиду редкости самого явления и отсутствия соответствующих корпусов текстов для русского языка. В данной работе осуществляется одна из первых попыток разрешения явления гэппинга для русского языка с использованием методов глубокого обучения в рамках соревнования AGRR-2019. Для определения наличия гэппинга в предложении мы применили подход на основе рекуррентной нейронной сети, а для полной аннотации использовали нейросетевую архитектуру Universal Transformer, основанную на механизме внимания с рекуррентными связями в глубину. Используя также предобученные векторные представления слов fastText, мы получили результат 85% (стандартная F-мера) для задачи бинарной классификации и 62% (посимвольная F-мера) — для задачи полной аннотации. Мы предполагаем, что фиксированные векторные представления слов как fastText не содержат достаточно синтаксической информации для корректного сопоставления «остатков» с их коррелятами в предложениях с гэппингом. Мы также видим основания считать, что наша модель имеет более высокую способность к генерализации, если не учитывать пунктуацию при обучении и проверке модели.

Ключевые слова: гэппинг, Universal Transformer, fastText, глубокое обучение, NLP

1. Introduction

According to [Ross, 1970], gapping is a type of ellipsis in which a repeated main verb is elided in one or more conjuncts of a coordinate structure, such as in the example (1).

(1) *Moj otec znal ego otca, moj ded — ego deda.*
My father knew his father, my grandfather — his grandfather.

Despite the fact that this phenomenon was widely discussed from a theoretical point of view, there is still no consensus on some cases. For example, gapping can occur in comparative constructions and "short" answers. Moreover, the differentiation between gapping and other types of ellipsis (such as VP-ellipsis and stripping) is not trivial. We refer to [Johnson, 2014] for more details.

However, from a practical point of view, it pose challenges as well. First of all, it is not obvious how a conjunct with the elided main verb should be presented in a sentence's dependency representation since all dependency representations consider a verb to be the head of a clause. Different approaches were proposed to address this issue, including adding empty nodes [Boguslavsky et al., 2002] and incorporating new or adapting existing dependency relations [Schuster et al., 2018].

The second problem is connected with syntactic parsers. Continuing with the example (1), one option is to reconstruct the verb *znal (knew)* in the sense of its wordform and linear position in the sentence. Then standard parsers should perform well, but such reconstruction is a challenging task itself. Another option is to develop a parser that correctly deals with a clause with a gap and then reconstructs the gap. It is possible not to reconstruct the gap in the latter approach, but indicating the elided material is essential for downstream tasks such as semantic role labeling [Matthew Lamm and Liang, 2018] and semantic parsing [Ge and Mooney, 2009]. The rarity of the phenomenon of ellipsis in natural languages [Droganova and Zeman, 2017] and lack of training data makes the latter approach even more difficult for statistical parsers.

It is worth to mention that there were a couple of attempts to address this phenomenon in Russian language from practical point of view. In a recent paper [Droganova et al., 2018] the authors trained two existing statistical parsers on a corpus pre-enriched with sentences with gapping. They obtained some improvements of the parsing accuracy of gapping in Russian compared to the baseline where the corpus was not enriched with the gapped sentences, but the improvement was not significant. Another attempt is presented in paper [Bogdanov, 2012] and is basically an extension to an existing rule-based parser. Unfortunately, the paper lacks any evaluation and the approach is strongly dependent on the parser.

In this paper, we propose an automatic gapping resolution system for Russian based on recently proposed Universal Transformer neural network architecture. Our model was evaluated during AGRR-2019 competition. The paper is structured as follows. In **Section 2** we give an overview of data and task description. Moreover, since there is no generally acceptable theoretical definition of gapping, we formulate a working definition based on the data provided. In **Section 3** we describe our approach in details. The results are presented in **Section 4** and the conclusion is provided in **Section 5**.

2. Data and task description and gapping definition

As was mentioned in **section 1**, the evaluation of the proposed model was performed during the AGRR-2019 competition. The organizers provided a corpus of several thousands of sentences from texts of different genres. The corpus statistics is shown in **Table 1**.

Table 1: AGRR-2019 corpus statistics

	Training set	Development set	Test set
Sentences with gapping	5,542	1,382	636
Sentences without gapping	10,864	2,760	1,409

Because some cases of gapping are controversial from theoretical point of view, automatic gapping resolution cannot be held in its entirety. So it is necessary to provide a working definition of gapping that, according to the data provided, can be formulated as follows.

Definition 1 (Working definition of gapping in Russian) *Gapping is a type of ellipsis in which a repeated finite verb, possibly along with contiguous portions of its verb phrase, is elided in one or more clauses conjoined to the right of a clause containing the same verb, with a remnant material at least to the right of the gap.*

Here, the remnant material is the contiguous overt material in a gapped clause. Since the elided material is contiguous, there are no more than two remnants in a gapped clause. Not only a main verb can be omitted in a gapped clasue of a sentence in Russian such as in the example (2), but there are not such examples in the data provided. But at the same time stripping and left node raising are considered as gapping.

(2) *On ee v stol položit, a my voz'mem da v škap preložim...*
He it in table put, and we to cupboard moved...
(He put it in his table, and we moved it to the cupboard...)
[M. E. Saltykov-Ŝedrin. Gospoda Golovlevy (1875–1880)]

Each sentence in the corpus is annotated as follows:

1. There is a label indicating whether a sentence contains a gap or not.
2. If there is a gapping construction in a sentence, character offsets for annotation borders for each gapping element are provided. Namely, these elements are: the elided predicate (V) with its remnants (R_1, R_2) for every gapped clause; the head of the correspondent predicate (cV) with the correlates of the remnants (cR_1, cR_2) for the initial conjunct.

So, since there is a gapping construction in the sentence from the example (1), the annotation will look as follows:

(3) [$_{cR_1}$*Moj otec*] [$_{cV}$*znal*] [$_{cR_2}$*ego otca*] , [$_{R_1}$*moj ded*] — [$_V$] [$_{R_2}$*ego deda*] .

Overall, we can classify (see also **Table 2**) gapping constructions presented in the corpus by:

1. type of gap:
   (a) single predicate;
   (b) predicate with portions of its verb phrase (contiguous material);
2. number of gapped clauses:
   (a) one clause;
   (b) more than one clause;

3. number of remnants:
   (a) one remnant;
   (b) two remnants.

Table 2: Extended AGRR-2019 corpus statistics. Only sentences with gapping are included. Number of sentences with different types of gap were estimated by distance between the head of the verb phrase in the initial conjunct and one of the correlates of the remnants.

	Training set	Development set	Test set
Type of gap			
Single predicate	4,581	1,141	583
Predicate-arguments	961	241	97
Number of gapped clauses			
One clause	5,173	1,292	632
More than one clause	369	90	48
Number of remnants			
One remnant	77	27	17
Two remnants	5,465	1,355	663

Three tasks were presented by the organizers.

1. Binary classification. For a given sentence decide if there is a gapping construction in it.
2. Gap resolution. Predict the position of the elided predicate and the correspondent predicate in the antecedent clause.
3. Full annotation. In each clause with the gap predict the linear position of the elided predicate and annotate its remnants. In the antecedent clause find the constituents that correspond the remnants and the predicate that corresponds the gap.

3. Model description

Recurrent and convolutional neural networks has shown promising results in natural language processing tasks in recent years [Yin et al., 2017], [Young et al., 2018]. Despite the fact that every hidden state update in RNN takes previous states into account, however, combining attention mechanism [Bahdanau et al., 2014] with RNNs has become a standard for solving different tasks, especially for encoder-decoder based machine translation systems [Wu et al., 2016]. It led to developing network architectures based solely on attention without any recurrence or convolution. One such model, the Transformer [Vaswani et al., 2017], has established new state-of-the-art results on machine translation tasks. However, one limitation of the network is that it does not generalize well to input lengths not encountered during training. To make the network computationally universal and, in particular, to overcome the mentioned issue, the Universal Transformer with recurrence over depth (unlike RNNs in which recurrence is over time) was recently proposed [Dehghani et al., 2018]. The latter model was also shown to be able to capture dependency structure of a sentence, outperforming the vanilla Transformer significantly. Since the gapping phenomenon is considered to be purely syntactic, we decided to use a part of the Universal Transformer network in our model.

Speaking about the task, the main observation is that the position for the elided predicate is known after we found the offsets for its remnants: it immediately precedes the second remnant (or the first if it is the only one) in each gapped clause. It implies that all three proposed tasks could be treated as one and it is now straightforward to formulate the joint task as a sequence labeling problem. Namely, the labels are $\{R_1, R_2, cV, cR_1, cR_2, nG\}$, where nG is a label for a word not connected with gapping phenomenon. Now, the example (3) transforms into

(4)	*Moj*	*otec*	*znal*	*ego*	*otca*	*,*	*moj*	*ded*	—	*ego*	*deda*	*.*
	cR_1	cR_1	cV	cR_2	cR_2	nG	R_1	R_1	nG	R_2	R_2	nG

Below we explain the models we evaluated to solve the task in more details. The code is publicly available on GitHub[1].

3.1. Data representation

Since the input data is raw text (splitted into sentences), some data preprocessing must be made. First of all, we tokenized the sentences, using NLTK library [Loper and Bird, 2002] with external tokenization model for Russian[2]. But the problem is that the model can generalize worse paying too much attention to punctuation. Even a simple binary classifier that predicts whether a sentence contains a gapping construction in it based solely on presence of a dash achieves precision and recall of about 70% on the training set. That is why the second option we tried is to just ignore all punctuation and treat every character sequence surrounded by non-alphanumeric characters as a word.

Secondly, we used extended fastText word embedding [Bojanowski et al., 2017] pretrained on Wikipedia and Common Crawl [Grave et al., 2018]. It is based on skip-gram model [Mikolov et al., 2013] but each word is represented as a bag of character n-grams along with the word itself. Incorporating the subword information has two important advantages connected with the task:

1. it captures morphological information, improving performance on syntactic tasks, especially for morphologically rich languages such as Russian;

2. the model can produce word vectors for out-of-vocabulary (OOV) words treating a word as a set of n-grams.

[1] https://github.com/Derise/agrr

[2] https://github.com/Mottl/ru_punkt

3.2. Model architecture

We tried two different models. Both models assign a label for each word in a sentence, but one of them is divided into two submodules: one is a binary classifier (solves task 1) and another one is a multi-class classifier trained only on gapped sentences. Multi-class classifiers in both models share the same architecture.

3.2.1. Binary classifier

We used a 2-layer bidirectional gated recurrent neural network (biGRU) [Cho et al., 2014] with dropout [Gal and Ghahramani, 2016]. The hidden state on the last time step of the second layer is an input to a fully-connected layer with sigmoid activation. The cost function is cross entropy.

We used Adam optimizer [Kingma and Ba, 2014] with learning rate $\alpha = 0.000625$. We did not change learning rate during training: instead we adopted the approach presented in [Smith et al., 2017]. Namely, if the validation loss after an epoch is not minimal compared to all previous losses, the batch size is doubled. The upper bound for the batch size is limited to the memory size.

3.2.2. Multi-class classifier

The mentioned above Universal Transformer architecture consists of encoder and decoder with the same basic structures. But since we formulated our task as a sequence labeling problem, no decoder is needed. So we applied a softmax layer directly after the output of the encoder. No changes were made to the encoder architecture compared to the original version; therefore we skip the detailed description of the encoder and refer to the original paper [Dehghani et al., 2018].

Whether the classifier is trained on all sentences from the training corpus or only gapped ones, the configurations are the same. The hidden size is the size of word embedding (300), the number of self-attention heads is 6 and the model depth is 12. Adam optimizer is applied with the same learning rate as for binary classification model. The Universal Transformer takes more memory to train compared to biGRU with the same number of parameters (because recurrence over time steps in RNN is not parallelizable) and we could not increase the batch size. Instead, we used step decay scheme: every 5 epochs the learning rate is decreased by a factor of 2 if loss did not improve.

4. Results and error analysis

The results are shown in **Table 3**. Two different models were evaluated:

- biGRU+UT: biGRU as a binary classifier is evaluated and then the Universal Transformer encoder as a multi-class classifier is applied to tag each word in sentences which were predicted as gapped ones.
- UT(joint): the Universal Transformer encoder were trained on all sentences. If there are no words with tags R_1, cR_1 or cV in a given sentence, it is considered as the one without gapping in it.

Table 3: Standard (for binary classification) and symbol-wise (for other tasks) F-measure of different models trained and evaluated on AGRR-2019 dataset

	Train			Dev			Test		
Model	**Binary**	**Gap**	**Full**	**Binary**	**Gap**	**Full**	**Binary**	**Gap**	**Full**
NLTK tokenizer									
biGRU+UT	0.97	0.82	0.78	**0.92**	**0.73**	**0.69**	**0.85**	**0.64**	**0.60**
UT(joint)	—	—	—	0.76	0.52	0.49	0.65	0.41	0.39
UT$_{LARGE}$(joint)	—	—	—	0.72	0.51	0.48	0.70	0.49	0.45
Simple tokenizer, no punctuation									
biGRU+UT	0.89	0.51	0.54	0.82	0.44	0.46	0.82	0.45	0.47
UT(joint)	0.70	0.41	0.41	0.66	0.34	0.34	0.65	0.35	0.34

Additionally, we tried bigger model of UT(joint) by adding fully connected layer before UT input, increasing the hidden size from 300 to 512 (increasing total parameters from 1.5M to 3M). Moreover, as was mentioned in section 3.1, two tokenization techniques were used: one is using NLTK library and another is a simple approach where all punctuation marks are ignored and each sentence is splitted into words by non-alphanumeric characters.

For binary classification task the metric is standard f-measure. For other tasks the metric is symbol-wise f-measure, here is the example from the organizers: “if the gold standard offset for certain gapping element is 10:15 and the prediction is 8:14, we have 4 true positive chars, 1 false negative char and 2 false positive chars and the resulting f-measure equals 0.727”.

Drawing attention to the biGRU+UT model, we take into consideration only those sentences which were correctly predicted by the binary classifier. Almost all *V* tags are correct (in terms of recall) and it turns out that the most challenging task for the model was remnants matching. The most frequent errors are:

1. Not determining an elision of dependents of a verb (5): here and throughout, the example (5a) is correct and the example (5b) is the output of the model.

(5) a. [$_{cR_1}$*Sverh”estestvennoe vmešatel’stvo v dela prirody*] [$_{cV}$*kazalos’*] *emu* [$_{cR_2}$*istočnikom užasa*] *, a* [$_{R_1}$*bessmertie*] *—* [$_{V}$] [$_{R_2}$*fatal’nym dlja nadeždy izbavit’sja ot boli*].

b. [$_{cR_1}$*Sverh”estestvennoe vmešatel’stvo v dela prirody*] [$_{cV}$*kazalos’*] [$_{cR_2}$*emu istočnikom užasa*] *, a* [$_{R_1}$*bessmertie*] *—* [$_{V}$] [$_{R_2}$*fatal’nym dlja nadeždy izbavit’sja ot boli*].

(The supernatural intervention in the affairs of nature seemed a source of horror to him, and the immortality—fatal for hope to get rid of pain.)

2. Incorrect remnants matching (6).

(6) a. *Hotja oni* [$_{cV}$*nazyvali*] [$_{cR_1}$*tebja*] [$_{cR_2}$*drugom*] *, a* [$_{R_1}$*ee*] [$_{V}$] [$_{R_2}$*podrugoj*]*!*

b. [$_{cR_1}$*Hotja oni*] [$_{cV}$*nazyvali*] [$_{cR_2}$*tebja drugom*] *, a* [$_{R_1}$*ee*] [$_{V}$] [$_{R_2}$*podrugoj*]*!*

(Although they called you a friend, and her a friend!)

3. Determining multiple gaps when there is only one (7).

(7) a. *V obŝem,* [$_{cR_1}$*politiki i generaly*] [$_{cV}$*pozabotilis’*] [$_{cR_2}$*o svoih mestah, o svoej kar’ere, o svoih kreslah*] *, a* [$_{R_1}$*kto-to eŝe*] [$_V$] [$_{R_2}$*i o svoih karmanah*].
b. [$_{cR_1}$*V obŝem,* [$_{cR_1}$*politiki i generaly*] [$_{cV}$*pozabotilis’*] [$_{cR_2}$*o svoih mestah*], [$_{cR_1}$*o svoej kar’ere*], [$_V$] [$_{cR_2}$*o svoih kreslah*]*, a* [$_{R_1}$*kto-to*] [$_V$] [$_{R_2}$*eŝe i o svoih karmanah*] .
(Well, the politicians and the generals have taken care about one’s places, one’s career, one’s armchairs, and somebody also about one’s pockets.)

To compare the results with other participants of the competition we used biGRU+UT model with NLTK tokenizer with minor hyperparameters tuning for the UT part. The comparative results (obtained for the test set) are shown in **Table 4**.

For this final model we also reviewed the results for the sentences with gapping in the test set according to the classification mentioned in **Section 2**. It is shown in **Table 5**.

The most conspicuous result is for full annotation of sentences with one remnant. The reason is that the model almost always finds two remnants (8).

(8) a. [$_{cV}$*Dobavljaem*] [$_{cR_1}$*muku, krahmal i razryhlitel’*] *, a v konce —* [$_V$] [$_{cR_1}$*smetanu*].
b. [$_{cV}$*Dobavljaem*] [$_{cR_1}$*muku, krahmal*] [$_{cR_2}$*i razryhlitel’*] *, a* [$_{cR_1}$*v konce*] — [$_V$] [$_{cR_2}$*smetanu*].
(Add flour, starch and baking powder, and sour cream at the end.)

Another interesting observation is that full annotation performance is weaker if the elided material includes portions of the VP along with the main verb itself. That is because these portions tend to become a part of remnants’ correlates (5), (9).

(9) a. [$_{cR_1}$*Ono*] [$_{cV}$*dolžno*] *zahvatit’* [$_{cR_2}$*vas*] *, a* [$_{cR_1}$*ne vy*] [$_V$] [$_{cR_2}$*ego*].
b. [$_{cR_1}$*Ono*] [$_{cV}$*dolžno*] [$_{cR_2}$*zahvatit’ vas*] *, a* [$_{cR_1}$*ne vy*] [$_V$] [$_{cR_2}$*ego*].
(It must capture you, and not you him.)

Table 4: The comparative AGRR-2019 results of competitors’ models. The best three results in each task are in bold.

Team	Binary			Gap resolution	Full
	Precision	Recall	F-measure	F-measure	F-measure
fit_predict	0.97	0.95	**0.96**	**0.90**	**0.89**
EXO	0.90	0.96	**0.93**	**0.81**	**0.79**
Koziev Ilya	0.78	0.90	0.83	**0.68**	**0.65**
Derise	0.80	0.91	**0.85**	0.66	0.62
Meanotek	0.89	0.78	0.83	0.64	0.51
МГУ-DeepPavlov	0.93	0.64	0.76	0.60	0.59
Vlad	0.78	0.92	0.84	0.57	—
MorphoBabushka	0.76	0.62	0.68	0.47	0.44
nsu-ai	0.49	0.126	0.20	0.04	0.04

Table 5: Extended results of the final model for the test set. Since the results are shown only for the sentences with gapping, precision and standard F-measure for binary classification are not shown.

	Binary	Gap resolution	Full
	Recall	F-measure	F-measure
Type of gap			
Single predicate	0.91	0.82	0.79
Predicate-arguments	0.89	0.80	0.69
Number of gapped clauses			
One clause	0.90	0.81	0.76
More than one clause	0.98	0.84	0.76
Number of remnants			
One remnant	0.76	0.69	0.24
Two remnants	0.91	0.82	0.78

With regard to binary classification the main observation is that the more information related to the phenomenon of gapping is presented in a sentence, the higher predictions are made by the classifier.

5. Conclusion

In this paper we proposed the approach to automatic gapping resolution in Russian. Looking at the results it is clear that:

1. RNN-based approach achieves reasonable performance on binary classification task.
2. Increasing the number of parameters in the Universal Transformer did not improve the results. One explanation is that fixed word embedding like fastText does not contain enough syntactic information. Probably, contextual representations such as ELMo [Peters et al., 2018] or BERT [Devlin et al., 2018] would perform better.
3. Remnants matching is the most challenging task for the model.
4. Our model is not capable to deal with sentences with gapping with one remnant, possibly in part owing to the small number of such examples presented in the corpus.
5. Our model generalize better if punctuation marks are removed. Unfortunately, it did not improve the overall performance since the fastText model was pretrained on texts that contain punctuation marks.

Acknowledgements

This work is supported by the Russian Science Foundation under grant 16-18-10422.

References

1. *Bahdanau, D. et al.:* Neural machine translation by jointly learning to align and translate. arXiv preprint arXiv:1409.0473. (2014).
2. *Bogdanov, A.:* Description of gapping in a system of automatic translation. In: Computational linguistics and intellectual technologies: Papers from the annual conference "dialogue". (2012).
3. *Boguslavsky, I. et al.:* Development of a dependency treebank for russian and its possible applications in nlp. In: Proceedings of the third international conference on language resources and evaluation (lrec-2002). pp. 852–856, Las Palmas (2002).
4. *Bojanowski, P. et al.:* Enriching word vectors with subword information. Transactions of the Association for Computational Linguistics. 5, 135–146 (2017).
5. *Cho, K. et al.:* Learning phrase representations using rnn encoder-decoder for statistical machine translation. arXiv preprint arXiv:1406.1078. (2014).
6. *Dehghani, M. et al.:* Universal transformers. arXiv preprint arXiv:1807.03819. (2018).
7. *Devlin, J. et al.:* BERT: pre-training of deep bidirectional transformers for language understanding. CoRR. abs/1810.04805, (2018).
8. *Droganova, K. et al.:* Mind the gap: Data enrichment in dependency parsing of elliptical constructions. In: Proceedings of the second workshop on universal dependencies (udw 2018). pp. 47–54 (2018).
9. *Droganova, K., Zeman, D.:* Elliptic constructions: Spotting patterns in ud treebanks. In: Proceedings of the nodalida 2017 workshop on universal dependencies (udw 2017). pp. 48–57 (2017).
10. *Gal, Y., Ghahramani, Z.:* A theoretically grounded application of dropout in recurrent neural networks. In: Advances in neural information processing systems. pp. 1019–1027 (2016).
11. *Ge, R., Mooney, R. J.:* Learning a compositional semantic parser using an existing syntactic parser. In: Proceedings of the joint conference of the 47th annual meeting of the acl and the 4th international joint conference on natural language processing of the afnlp: Volume 2-volume 2. pp. 611–619 Association for Computational Linguistics (2009).
12. *Grave, E. et al.:* Learning word vectors for 157 languages. In: Proceedings of the international conference on language resources and evaluation (lrec 2018). (2018).
13. *Johnson, K.:* Gapping, (2014).
14. *Kingma, D. P., Ba, J.:* Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980. (2014).
15. *Loper, E., Bird, S.:* NLTK: The natural language toolkit. arXiv preprint cs/0205028. (2002).
16. *Matthew Lamm, D. J., Arun Chaganty, Liang, P.:* QSRL: A semantic role-labeling schema for quantitative facts. In: El-Haj, M. et al. (eds.) Proceedings of the eleventh international conference on language resources and evaluation (lrec 2018). European Language Resources Association (ELRA), Paris, France.
17. *Mikolov, T. et al.:* Distributed representations of words and phrases and their compositionality. In: Advances in neural information processing systems. pp. 3111–3119 (2013).

18. *Peters, M. E. et al.:* Deep contextualized word representations. arXiv preprint arXiv:1802.05365. (2018).
19. *Ross, J. R.:* Gapping and the order of constituents. Progress in linguistics: A collection of papers. 43, 249–259 (1970).
20. *Schuster, S. et al.:* Sentences with gapping: Parsing and reconstructing elided predicates. arXiv preprint arXiv:1804.06922. (2018).
21. *Smith, S. L. et al.:* Don't decay the learning rate, increase the batch size. arXiv preprint arXiv:1711.00489. (2017).
22. *Vaswani, A. et al.:* Attention is all you need. In: Advances in neural information processing systems. pp. 5998–6008 (2017).
23. *Wu, Y. et al.:* Google's neural machine translation system: Bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144. (2016).
24. *Yin, W. et al.:* Comparative study of cnn and rnn for natural language processing. arXiv preprint arXiv:1702.01923. (2017).
25. *Young, T. et al.:* Recent trends in deep learning based natural language processing. IEEE Computational Intelligence Magazine. 13, 3, 55–75 (2018).