Proceedings of the 13th Conference on Language Resources and Evaluation (LREC 2022), pages 7140–7146
Marseille, 20-25 June 2022

HerBERT Based Language Model Detects Quantifiers and Their Semantic Properties in Polish

Marcin Woliński, Bartłomiej Nitoń, Witold Kieraś, Jakub Szymanik
Institute of Computer Science Polish Academy of Sciences
Institute for Logic, Language, and Computation, University of Amsterdam
{marcin.wolinski, bartlomiej.niton, witold.kieras}@ipipan.waw.pl, j.k.szymanik@uva.nl

Abstract

The paper presents a tool for automatic marking up of quantifying expressions, their semantic features, and scopes. We explore the idea of using a BERT based neural model for the task (in this case HerBERT, a model trained specifically for Polish, is used). The tool is trained on a recent manually annotated Corpus of Polish Quantificational Expressions (Szymanik and Kieraś, 2022). We discuss how it performs against human annotation and present results of automatic annotation of 300 million sub-corpus of National Corpus of Polish. Our results show that language models can effectively recognise semantic category of quantification as well as identify key semantic properties of quantifiers, like monotonicity. Furthermore, the algorithm we have developed can be used for building semantically annotated quantifier corpora for other languages.

Keywords: quantifiers, annotation, HerBERT

1. Introduction

Quantifying expressions or quantifiers are understood in this text as language expressions which indicate quantity. These can be exact numbers of objects ('three [books]') or generalised quantifiers: 'all', 'each', 'some', 'none', 'majority [of]'. Quantifiers can also count events or express their frequency: 'never', 'always', 'twice', 'often', 'repeatedly', 'each Tuesday' (Szymanik and Kieraś, 2022).

The task of detecting quantifiers in the text is superficially similar to named entity recognition. The set of possible quantifiers comprises numerals expressed with digits and a seemingly limited set of other phrases, which, with some effort, could be listed in a dictionary. However, unlike in the case of named entities, not all occurrences of these expressions are in fact quantifiers. For example the digit '3' denotes a quantifier in the contexts where it is read as 'three' but not when it means 'third'. Similarly, nouns naming numbers, e.g. Polish *trójka*, may express a similar concept as numerals (*trójka chłopców* 'three boys') or it may denote any object bearing the number 3 (a radio station, TV channel, military unit, squad). In this latter context the word is not a quantifier. For another example, consider the word *most*. It can denote a quantifier when followed by a plural noun as in *most men* or a modifier when followed by an adjective as in *most beautiful*. Thus, the decision whether a given expression is a quantifier is to a large extent based on the *semantic analysis* of the linguistic context in which the expression appears. Beyond detecting quantifiers, we are also interested in two subordinate tasks: having a quantifier we want to characterise its semantic features (e.g., monotonicity) and to determine its scope, that is to find in the text the expression specifying what is being quantified.

Furthermore, by showing that natural language model based on BERT can recognise quantifiers and their semantic features, we also contribute to understanding of semantic knowledge of language models, see, e.g., (Linzen et al., 2018; Linzen et al., 2019).

2. The Corpus and its Annotation Scheme

Although the use of quantifying expressions in natural languages has been studied in linguistics and philosophical logic for many decades now (Peters and Westerståhl, 2006; Szymanik, 2016), a corpus linguistic and computational approach to the problem is relatively new. Very few resources exist for such research, in particular large datasets suitable for machine learning methods are scarce. According to our knowledge, the only large-scale attempt at the manual annotation of generalized quantifiers was made for Polish (Szymanik and Kieraś, 2022), thus in our paper we use the Corpus of Polish Quantificational Expressions (CPQE) as the only manually annotated data set. Recently, there has been an effort to establish an ISO standard annotation scheme for quantification phenomena in natural language as part of the ISO Semantic Annotation Framework (ISO 24617) (Bunt, 2020), which could be seen as a growing interest in the field promising more development to be conducted and data sets to be published in the near future. However, the developments of the ISO standard are still very much in the preliminary stage and the first proposal of the standard was published after CPQE annotation was concluded.

Technically, the corpus of Polish quantifying expressions was created by adding a new annotation layer to the gold-standard 1.2 million tokens large subcorpus of the National Corpus of Polish (Przepiórkowski et al., 2012). This corpus comprises a balanced set of short samples (approx. 40-60 words long up to full sentences) representing different text genres. The corpus of quantifiers is available on the web both as a separate layer of annotation together with the whole NKJP1M indexed in the corpus search engine as well as an XML source

(1) *Za Nowym Ładem zagłosowało* ***dwudziestu trzech*** ⟨*decydentów*⟩, *przeciw było* ***437***,
for New Deal voted twenty three decision-makers, against were 437,
D:num:nmon:nmon D:num:nmon:nmon
nikt *się nie wstrzymał.*
nobody REFL not abstained.
D:exst:dec:dec
'Twenty-three decision makers voted for the New Deal, 437 were against, none abstained.'

(2) *Dziennikarze* ***wielokrotnie*** ⟨*informowali*⟩ *o nieprawidłowościach.*
Journalists repeatedly reported about irregularities.
A:exst:nmon:inc
'Journalists have repeatedly reported on the irregularities.'

Figure 1: Quantifier annotation in CPQE. Quantifiers (in bold) are annotated with features discussed in the text. Angle brackets mark quantifier scopes. The word *decision-makers* is the common scope for quantifiers *twenty-three* and *437*. The quantifier *none* has no scope.

tarball (Szymanik and Kieraś, 2022)[1]. The latter form, encoded according to TEI P5, was used for training neural models presented in this article.
In CPQE, two main types of quantifiers are distinguished (see above for an example):

- D-quantifiers are a part of dependant expressions in the predicate argument structure of sentences. Typically, they are determiners in nominal phrases.
- A-quantifiers directly build or modify predicates. Typically, these are adverbs modifying verbs.

The manually introduced annotation in the corpus consists of three elements. First, the quantifier itself is marked in the text (as a continuous sequence of words). Second, it is described using the following features:

- Type: D-quantifiers and A-quantifiers (see above)
- Subtype distinguishes between existential (intersective), e.g., *some*, universal (co-intersective), e.g., *all*, proportional, e.g., *many*, and numeral quantifiers (exact amounts), e.g., *5*; see (Keenan and Paperno, 2017) for the exact definition.
- Monotonicity describes quantifier's monotonicity, left and right monotonicity is annotated as two separate features but with the same range of values: increasing, decreasing, and non-monotonic. See (Peters and Westerståhl, 2006) for the definition.

Finally, a scope is assigned to the quantifier, showing what is being quantified. By a convention, a maximal nominal phrase is marked as the scope of a D-quantifier and a full verbal form (including potential negation particles, reflexive markers, and auxiliaries) as the scope of an A-quantifier. The scope can be empty, as it may be omitted in the text. The scope may also be shared by more than one quantifier.

[1] http://kwantyfikatory.nlp.ipipan.waw.pl

Detailed information about considered quantifiers, their features, and rules of annotation can be found in the paper (Szymanik and Kieraś, 2022).

3. The Proposed Method

The process of automatic marking up quantifiers can be divided into three tasks: detecting quantifiers, classifying quantifiers, and detecting their scopes. Detection of quantifiers means deciding that some continuous spans of the text form quantifying expressions. Classification involves labels consisting of 4 parts with theoretical number of $2 \times 4 \times 3 \times 3 = 72$ combinations but in practice 58 labels are used. The quantifier scope is again a continuous span of text, but in this case a quantifier, for which the scope is sought, needs to be specified as part of the input data.

In our first attempt at quantifier detection, we've tried methods successfully used for named entity recognition. Conditional Random Fields (Sutton and McCallum, 2012) are used for example in Liner2 tool (Marcińczuk and Janicki, 2012). We used a wide range of hand crafted features in our experiments: local features like base form, part-of-speech information and those taken from dependency trees created using COMBO tool (Rybak and Wróblewska, 2018). The best we've achieved for quantifier detection was about 0.8258 F1 score. If quantifier features are included, the F1 score drops to less than 0.7. The disadvantage of this method, apart from the relatively low scores, is using hand crafted features which need to be provided by external tools and can be language dependent.

In the work reported in this paper, we decided to use modern models based on neural networks, in particular on the Transformer architecture and BERT. Under this methodology a pre-trained BERT model is extended with new layers specific to the task at hand. As the pre-trained model we have selected HerBERT (Mroczkowski et al., 2021), which is the best BERT-type model available for the Polish language according

Input		Output
Za</w>	→	O
Nowym</w>	→	O
Ład	→	O
em</w>	→	O
za	→	O
głosowało</w>	→	O
dwudziestu</w>	→	B
trzech</w>	→	I
decyden	→	O
tów</w>	→	O
,</w>	→	O
nikt</w>	→	B
się</w>	→	O
nie</w>	→	O
wstrzymał</w>	→	O
.</w>	→	O

Figure 2: Detection of quantifiers as a token labelling task. Tokens for the sentence *Za Nowym Ładem zagłosowało dwudziestu trzech decydentów, nikt się nie wstrzymał.* 'Twenty-three decision-makers voted for the New Deal, none abstained.' as generated by HerBERT tokenizer.

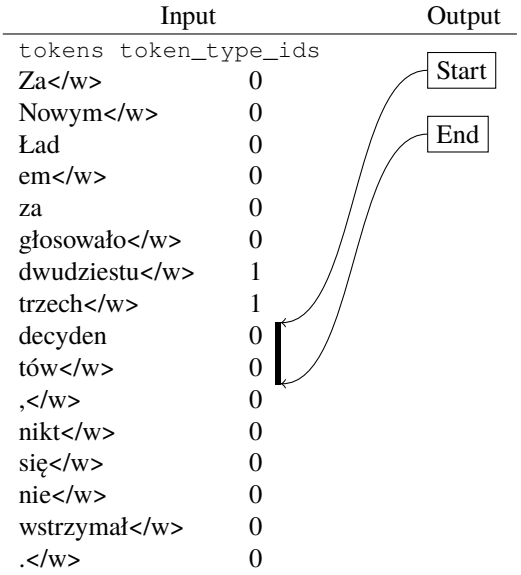

Figure 3: Detection of quantifier scopes as an instance of extractive question answering. In the example, the model is asked to find the scope for the quantifier *dwudziestu trzech* 'twenty three'. The expected answer is the word *decydentów* 'decision-makers' (which was split into 2 tokens by HerBERT's tokenizer)

to the average score on KLEJ benchmark tasks[2] (Rybak et al., 2020). We have decided to use „base" variant of the model, which is a compromise between necessary computing power (in particular GPU memory) and quality. It is worth noting that the "base" version of HerBERT outperforms some other "large" models presented on the KLEJ leaderbord.

The experiments were performed using Huggingface Transformers implementation in the Torch version. The detection procedure is performed on the text divided into individual sentences.

3.1. Detection of Quantifying Expressions

Quantifying expressions are specific spans of tokens in the text. Detection of quantifiers can be expressed as a token labelling task, where each token is classified as belonging or not belonging to a quantifier expression. According to the annotation rules of CPQE, a quantifier cannot be nested in another quantifier. Thus, a simple IOB markup can be used to represent the quantifiers (B marks the first token of a quantifier, I ("inside") marks non-first tokens, O is "outside"), as illustrated in Figure 2.

This type of processing can be performed by Huggingface model of type BertForTokenClassification. The model consists of the selected pre-trained BERT layer and a dense layer that performs classification, trained from scratch.

[2] https://klejbenchmark.com/leaderboard/

3.2. Assignment of Quantifier Features

Deciding features of a given quantifier can be described as a sequence classification task. In the context of BERT based methods, this task can be solved in a rather straightforward manner with the model BertForSequenceClassification: the quantifier is given as input and a label is produced as output.

As noted before, the labels used in CPQE consist of four separate features. Thus, two options arise: either the labels are predicted as atomic values or each of the features is predicted independently. In the latter case a variant of the model is needed, where the common BERT layer is extended with four independent dense layers predicting respective features. We hoped that independent prediction would perform better, since the distribution of individual features is less skewed than the distribution of whole labels. However, both approaches showed very similar results. In consequence, we have decided to predict whole labels, since this model is conceptually simpler.

3.3. Detection of Quantifier Scopes

Quantifier scope is a span of tokens in the same sentence as quantifier expression. We model detection of scopes as an instance of extractive Question Answering, where the "question" is the quantifier and the extracted answer is its scope. Since multiple quantifiers can appear in a single sentence, a particular quantifier has to be explicitly pointed at in the input data.

This type of processing can be performed using Huggingface model named BertForQuestionAnswering. Typically, the input tokens for this model include the

question, a special separator token, and then the text from which the answer is to be extracted. Besides being marked with a separator, the distinction between question and text is expressed with an array of numbers (`token_type_ids`) containing 0s for tokens belonging to the question and 1s for tokens in the text. Thus, each input position fed to BERT comprises a token id and a type id.

The above representation is not perfect for scope detection. It happens sometimes that a sentence contains multiple occurrences of the same quantifier with different scopes (e.g. *3 apples and 3 oranges*). In this case we need to point the model to one of the occurrences to get the right scope in result.

We propose a representation where the input is only the sentence being processed and one of quantifiers in that sentence is marked with `token_type_ids` (Figure 3). As it turns out, although this representation is a bit different from what the model's authors assumed, BERT is able to successfully learn from such data.

In this model, a dense layer, placed on top of a pre-trained BERT layer, produces two numerical pointers expressing the start and end positions of the quantifier scope in the sentence.

Each of the tasks described in the last three subsections is solved using a pre-trained BERT model with respective classification layer added. However, since the tasks belong to different types, it would be difficult to train a joint model. Thus, we have decided to train the models separately.

4. Evaluation

4.1. Cross-validation

To fully utilize the available data for validation, we have performed a 5-fold cross-validation on the data of The Corpus of Quantificational Expressions (as of January 2021). Tables 1 and 2 present the results. Only documents with at least one quantifier were taken to the evaluation data. Folds were created so as to contain possibly same number of quantificational expressions using the following procedure:

- sort documents by number of quantifiers in the descending order,
- for each document in the list, add it to the fold which is presently smallest in terms of the number of quantifiers.

The procedure resulted in three folds with 4388 quantifiers and two folds with 4387 quantifiers.

Presented scores were acquired using a model trained with following hyperparameters (determined in experiments on a development set):

- 2 epochs of HerBERT model fine tuning, with 32 batch size and 1.5e-5 learning rate for quantifiers prediction (IOB);
- 8 epochs of HerBERT model fine tuning, with 64 batch size and 3e-5 learning rate for scopes prediction;
- 4 epochs of HerBERT model fine tuning, with 64 batch size and 3e-5 learning rate for features prediction (Tags).

Table 1 presents the results of quantifiers detection. Scores for „IOB" rows were counted using seqeval library[3]. „IOB only" row present score for predicting only quantifier tokens, while „IOB + Tags" row present scores acquired when Tags are counted on system predicted quantifiers not on the gold ones (which in contrary is presented in the table 2). Last row of table 1 presents scopes detection accuracy counted on the whole token spans and predictions made using gold quantifiers.

Table 2 presents scores acquired for predicting complete quantifier feature set („complete tag") and each feature separately. The predictions were made using gold quantifiers as input.

As we can see, scores are quite satisfying for all prediction subtasks.

	Precision	**Recall**	**F1**
IOB only	0.8590	0.9068	0.8823
IOB + Tags	0.7968	0.8411	0.8183
		Accuracy	
Scopes		0.9013	

Table 1: Quantifiers and scopes detection scores. Values in *IOB + Tags* row counted using micro averaging.

	Precision	**Recall**	**F1**
Complete Tag	0.9177	0.9177	0.9177
Type	0.9948	0.9948	0.9948
Subtype	0.9613	0.9613	0.9613
Monoton_l	0.9540	0.9540	0.9540
Monoton_r	0.9561	0.9561	0.9561

Table 2: Quantifier features detection scores. Values counted using micro averaging.

When we compare table 2 and 3 we can see that for trained model as for human annotators the hardest feature to predict is monotonicity, while the easiest feature to predict is quantifier type.

Type	0.90
Subtype	0.76
Monoton_l	0.62
Monoton_r	0.63

Table 3: Inter-annotator Agreement (Cohen's Kappa) for separate features of quantifiers based on tokens annotated by two annotators (Szymanik and Kieraś, 2022)

[3]https://github.com/chakki-works/seqeval

4.2. The Model vs Humans

To get a better understanding of how the model performs against human annotators, we have taken a look at a random sample of differences. We have compared the annotation produced by the model in all folds of cross-validation with respective gold-standard files. From the set of differences (quantifiers missing, marked differently or with different scope), a random sample of 150 was drawn. These differences were assessed by the authors with results shown in Table 4.

Annotator was right	28%
System was right	50%
Both wrong	7%
Partially correct	15%

Table 4: Who is right when the model differs form humans in annotation (on a random sample of 150 differences)

The system was right in the impressive 50% of differences. In 72% of these cases (i.e. in 36% of all differences) the system has spotted a quantifier that was missing in the gold-standard annotation. In 7% of differences both versions of annotation were plainly wrong, while in 15% some parts were better but other worse than in the opposing answer, producing a mixed assesment.

Our general feeling is that the model has definitely gained a human-like competence in marking quantifiers. The differences mostly show up in places where the decision is not obvious. It is also plain that the complete list of differences is a great indication of spots in the gold-standard corpus that are potential errors or which are tagged in an inconsistent manner.

An interesting question may also be whether the model simply memorises the list of quantifiers seen in the training data. The answer is clearly negative. It is most obvious with numbers written in digits: the model has clearly gained the insight that any such formation may be a quantifier. But there are also examples of more interesting quantifiers which were recognised by the system although they did not appear in the training data. For example in the following sentence the program has detected the noun *mnogość* 'multitude' and correctly marked a long coordinated nominal phrase as its scope:

(3) ***Mnogość*** ⟨*przysyłanych do redakcji*
multitude sent to editors
wakacyjnych kartek i pozdrowień⟩
holiday postcards and greetings
świadczy, że Czytelnicy KOTA w
proves that readers KOT during
letnią kanikułę poznawali najdalsze
summer dog-days explored farthest
zakątki świata.
corners world
'The multitude of holiday cards and greetings sent to the editors proves that the readers of KOT explored the farthest corners of the world during the hot summer.'

5. The NKJP Annotation

The approach presented in the paper was used to automatically annotate quantifying expressions in the large reference corpus, namely the 300 million tokens large National Corpus of Polish[4]. The quantifiers were indexed along with other layers of annotation (morphosyntax, dependency syntax, named entities) in the web-based search engine (Brouwer et al., 2017). Users may query for any instance of a quantifying expression in the corpus with respect to its orthographic form and quantifier features as well as other linguistic information contained in other layers of annotation. For example, one can query for all D-type existential quantifiers containing root element of the dependency tree[5]. Due to technical limitations of the search engine only quantifiers and their features were indexed, while the quantifiers' scopes were omitted and cannot be queried.

Table 5 presents percentage of quantifier values according to the annotation scheme in the automatically annotated 300 million tokens large balanced National Corpus of Polish divided into main genres distinguished in the corpus. Each cell shows the percentage of quantifiers assigned a certain feature's value in a certain genre, e.g. in non-fiction 85.82% of recognized quantifiers are D-quantifiers and only 14.18% of them are A-quantifiers. The last two rows present a total percentege for each value in the automatically annotated balanced corpus (NKJP300) compared with the manually annotated tranining data (NKJP1M). First of all it is interesting to note that the values in those two last rows are almost the same, which shows that the distribution of the feature values are similar. As expected, one of the most large groups among the quantifiers is unmodified numerals (29.82% in training data and 30.00% in the automatically annotated corpus). D-quantifiers, including the unmodified numerals are almost ten times more frequent than A-quantifiers. Existential quantifiers (34.01% and 32.62% respectively) are more frequent than universal ones (17.79% and 16.98%), which are again slightly more frequent than proportional (18.38% and 20.41%). These frequency numbers are in line with the semantic complexity predictions for a restricted domain of 36 English quantifiers described by (Szymanik and Thorne, 2017) (see also discussion in (Szymanik and Kieraś, 2022)).

6. Conclusions

The main contributions of this paper are twofold. First, an algorithm has been proposed which exhibits a human-like performance in marking quantifying expressions in the text. This shows that the knowledge

[4] nkjp.nlp.ipipan.waw.pl/

[5] The query for such quantifiers would look like this: `<q="D:exst:.*" /> containing [deprel="root"]`

Text type	A	D	exst	univ	num	prop
non-fiction literature	14,18%	85,82%	38,06%	21,74%	21,56%	18,64%
instructive & guidebooks	14,27%	85,73%	29,06%	19,90%	28,09%	22,96%
conversational	10,23%	89,77%	31,64%	23,64%	37,08%	7,65%
letters	11,69%	88,31%	36,07%	25,68%	25,06%	13,19%
fiction	13,73%	86,27%	43,50%	24,42%	18,53%	13,56%
academic writing & textbooks	13,65%	86,35%	27,12%	15,48%	24,79%	32,61%
interactive	13,66%	86,34%	39,38%	24,08%	20,58%	15,97%
static WWW pages	7,64%	92,36%	30,85%	16,32%	31,24%	21,59%
unclassified non-fiction book	14,96%	85,04%	36,54%	30,05%	15,78%	17,63%
journalism	7,97%	92,03%	29,36%	14,25%	34,27%	22,11%
quasi-spoken	6,68%	93,32%	34,08%	14,78%	32,14%	19,00%
legal and official	3,07%	96,93%	28,68%	15,63%	35,95%	19,75%
NKJP300M	**9,64%**	**90,36%**	**32,62%**	**16,98%**	**30,00%**	**20,41%**
NKJP1M	**9,74%**	**90,26%**	**34,01%**	**17,79%**	**29,82%**	**18,38%**

Text type	lmon inc	lmon dec	lmon nmon	rmon inc	rmon dec	rmon nmon
non-fiction literature	14,64%	35,43%	49,92%	52,31%	15,74%	31,94%
instructive & guidebooks	12,81%	27,59%	59,59%	51,64%	9,98%	38,38%
conversational	14,12%	37,05%	48,83%	40,99%	15,02%	43,99%
letters	11,14%	41,26%	47,60%	47,76%	17,44%	34,79%
fiction	14,72%	44,44%	40,84%	50,03%	21,70%	28,27%
academic writing & textbooks	12,48%	22,03%	65,49%	50,52%	9,28%	40,19%
interactive	16,47%	39,93%	43,60%	51,52%	18,78%	29,71%
static WWW pages	12,94%	25,04%	62,02%	44,06%	10,74%	45,19%
unclassified non-fiction book	15,36%	45,60%	39,04%	60,49%	18,03%	21,48%
journalism	9,94%	22,93%	67,13%	38,98%	10,75%	50,27%
quasi-spoken	15,87%	24,87%	59,26%	44,51%	11,72%	43,77%
legal and official	11,85%	28,39%	59,76%	42,53%	13,57%	43,90%
NKJP300M	**11,90%**	**27,96%**	**60,14%**	**43,28%**	**13,00%**	**43,72%**
NKJP1M	**12,52%**	**29,00%**	**58,47%**	**42,45%**	**13,14%**	**44,42%**

Table 5: Percentage of quantifier categories values in automatically annotated 300 million balanced NKJP corpus, presented by types of texts and in total. Counts in each category sum up to 100%.

embedded in language models provides enough hints for detecting such abstract notions like quantifier, scope, or monotonicity.

A novel element in the algorithm is the task representation used for detecting quantifier scopes (Section 3.3.). Experiments were conducted with Polish data, but no element of the algorithm is language dependent. With appropriate data the results should scale to other languages.

Second, a large corpus NKJP300 has been processed using the algorithm and the results were made available via a corpus search engine. While due to copyright reasons we are not able to release the annotated texts, we hope that corpus searches will enable interesting research on quantifiers in language.

The evaluation of algorithm's performance against humans shows the necessary direction of follow-up work. The first step should be to correct gold standard data by comparing it with results of the system. We are convinced that the system already is more consistent than humans, in particular it is less prone to overlooking quantifiers in the text. Only after such cleanup it may make sense to return to fine-tuning the parameters of the algorithm. Also, further experiments with the model should be performed to understand how the model represents and detects the semantic properties, see, e.g., (Jumelet et al., 2021).

Acknowledgements

The work being reported was financed by National Science Centre grant 2017/25/B/HS1/02911.

7. Bibliographical References

Brouwer, M., Brugman, H., and Kemps-Snijders, M. (2017). MTAS: A Solr/Lucene based Multi Tier Annotation Search solution. In *Selected papers from the CLARIN Annual Conference 2016, Aix-en-Provence, 26–28 October 2016, CLARIN Common Language Resources and Technology Infrastructure*, number 136, pages 19–37. Linköping University Electronic Press, Linköpings universitet.

Bunt, H. (2020). Annotation of quantification: The current state of ISO 24617-12. In *16th Joint ACL - ISO Workshop on Interoperable Semantic Annotation PROCEEDINGS*, pages 1–12, Marseille, May. European Language Resources Association.

Jumelet, J., Denić, M., Szymanik, J., Hupkes, D., and Steinert-Threlkeld, S. (2021). Language models use monotonicity to assess npi licensing. *Findings of the Association of Computational Linguistics*.

Keenan, E. and Paperno, D. (2017). *Handbook of Quantifiers in Natural Language*, volume 2. Springer.

Linzen, T., Chrupała, G., and Alishahi, A. (2018). Proceedings of the 2018 EMNLP workshop BlackboxNLP: Analyzing and interpreting neural networks for NLP. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*.

Tal Linzen, et al., editors. (2019). *Proceedings of the 2019 ACL Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, Florence, Italy, August. Association for Computational Linguistics.

Marcińczuk, M. and Janicki, M. (2012). Optimizing CRF-based model for proper name recognition in Polish texts. In A. Gelbukh, editor, *CICLing 2012, Part I*, volume 7181 of *Lecture Notes in Computer Science (LNCS)*, pages 258–269. Springer, Heidelberg, March.

Mroczkowski, R., Rybak, P., Wróblewska, A., and Gawlik, I. (2021). Herbert: Efficiently pretrained transformer-based language model for Polish.

Peters, S. and Westerståhl, D. (2006). *Quantifiers in Language and Logic*. Clarendon Press, Oxford.

Adam Przepiórkowski, et al., editors. (2012). *Narodowy Korpus Języka Polskiego*. Wydawnictwo Naukowe PWN, Warsaw.

Rybak, P. and Wróblewska, A. (2018). Semi-supervised neural system for tagging, parsing and lematization. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pages 45–54, Brussels, Belgium, October. Association for Computational Linguistics.

Rybak, P., Mroczkowski, R., Tracz, J., and Gawlik, I. (2020). KLEJ: Comprehensive benchmark for Polish language understanding.

Sutton, C. and McCallum, A. (2012). An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4(4):267–373, April.

Szymanik, J. and Kieraś, W. (2022). The semantically annotated corpus of Polish quantificational expressions. *Language Resources and Evaluation*. https://doi.org/10.1007/s10579-022-09578-4.

Szymanik, J. and Thorne, C. (2017). Exploring the relation of semantic complexity and quantifier distribution in large corpora. *Language Sciences*.

Szymanik, J. (2016). *Quantifiers and Cognition. Logical and Computational Perspectives*. Studies in Linguistics and Philosophy. Springer.

8. Language Resource References

NKJP Consortium. (2012). *National Corpus of Polish*. Institute of Computer Science, Polish Academy of Sciences.

Szymanik, Jakub and Kieraś, Witold. (2021). *Corpus of Polish Quantificational Expressions*. Institute of Computer Science, Polish Academy of Sciences.