Language Sciences 60 (2017) 80–93

Contents lists available at ScienceDirect

Language Sciences

journal homepage: www.elsevier.com/locate/langsci

Exploring the relation between semantic complexity and quantifier distribution in large corpora

Jakub Szymanik[a,*], Camilo Thorne[b]

[a] Institute for Logic, Language and Computation, University of Amsterdam, P.O. Box 94242, 1090 GE Amsterdam, The Netherlands
[b] Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart, Pfaffenwaldring 5b, D-70569 Stuttgart, Germany

ARTICLE INFO

Article history:
Available online 7 March 2017

Keywords:
Generalized quantifiers
Semantic complexity
Corpus analysis
Generalized linear regression models
Analysis of deviance

ABSTRACT

In this paper we study if semantic complexity can influence the distribution of generalized quantifiers in a large English corpus derived from Wikipedia. We consider the minimal computational device recognizing a generalized quantifier as the core measure of its semantic complexity. We regard quantifiers that belong to three increasingly more complex classes: Aristotelian (recognizable by 2-state acyclic finite automata), counting ($k+2$-state finite automata), and proportional quantifiers (pushdown automata). Using regression analysis we show that semantic complexity is a statistically significant factor explaining 27.29% of frequency variation. We compare this impact to that of other known sources of complexity, both semantic (quantifier monotonicity and the comparative/superlative distinction) and superficial (e.g., the length of quantifier surface forms). In general, we observe that the more complex a quantifier, the less frequent it is.

1. Introduction

Linguists and philosophers have been searching for various ways to estimate the complexity and expressiveness of natural language. One important debate pivots around the Equivalent Complexity Thesis (see Miestamo et al., 2008); that is, the question whether all languages of the world are equally complex or can express equally complex concepts. It is not surprising that such questions can sparkle lively discussion, after all, a proper answer would involve integrating many aspects of linguistics, e.g., grammatical complexity, cognitive difficulty, cultural diversity, etc. As Sampson et al. (2009) puts it:

> Linguists and non-linguists alike agree in seeing human language as the clearest mirror we have of the activities of the human mind, and as a specially important of human culture, because it underpins most of the other components. Thus, if there is serious disagreement about whether language complexity is a universal constant or an evolving variable, that is surely a question which merits careful scrutiny. There cannot be many current topics of academic debate which have greater general human importance than this one.

These endeavors are usually driven by different (but often related) questions: What are the semantic bounds of natural languages or, in other words, what is the conceptual expressiveness of natural language (see, e.g., Szymanik, 2016)? What is the 'natural class of concepts' expressible in a given language and how to delimit it (see, e.g., Barwise and Cooper, 1981; Piantadosi,

[*] Corresponding author.
E-mail addresses: J.K.Szymanik@uva.nl (J. Szymanik), camilo.thorne@ims.uni-stuttgart.de (C. Thorne).

http://dx.doi.org/10.1016/j.langsci.2017.01.006
0388-0001/© 2017 Elsevier Ltd. All rights reserved.

2011)? Are there differences between various languages with respect to semantic complexity (see, e.g., Everett, 2005)? Or from a more methodological perspective: how powerful must be our linguistic theories in order to minimally describe semantic phenomena (see, e.g., Ristad, 1993)? A similar question can be also asked from a cognitive angle: are some natural language concepts harder to process for humans than others (see, e.g., Feldman, 2000; Szymanik and Zajenkowski, 2010)?

The outcomes of such debates heavily depend on the underlying operationalization of the complexity notion, hence, we propose a measure of semantic complexity (see Szymanik, 2016). It focuses on the meaning of the quantifiers abstracting away from many grammatical details as opposed to, for example, typological (cf. McWhorter, 2001) or information-theoretic approaches (cf. Juola, 1998) known from the literature. The goal of this paper is to give a proof-of-concept that such an abstract notion of semantic complexity can be used (together with other linguistic factors) to explain or predict the distribution of quantifiers in natural language textual data.

In order to contribute to the above outlined debate we focus on one aspect of natural language: its ability to express quantities by using the wide repertoire of quantifier expressions, like ‘most’, ‘at least five’, or ‘all’ (see, e.g., Keenan and Paperno, 2012). We restrict ourselves to study generalized quantifiers (GQs) as described by Barwise and Cooper (1981), plus some of the counting and proportional quantifier forms[1] from Szymanik (2016), see Table 1. We identify their main surface forms or lexical variants with high precision, rather than trying to cover all—a considerable challenge given the size of our corpora and thus beyond the scope of this paper. In general, we observe the following:

Table 1
Quantifiers and their semantic complexity. Note: we assume *few* to be the dual of *most*, see also footnote 1. Also, we distinguish between *>1/2*, *<1/2* and other proportional quantifiers given that they constitute, arguably, the most common such quantifiers.

Class	Examples	Quantifier	Complexity
Aristotelian	‘every’, ‘some’	All, some	2-State acyclic FA
Counting	‘more than 4’, ‘at most 5’	>k, <k	$k+2$-State FA
Proportional	‘most’, ‘less than half’	Most, <1/2,	
	‘few’, ‘more than half’	few, >1/2,	PDA
	‘less than three-fifths’	<p/k,	
	‘more than two-thirds’	>p/k	

(H) There is a relation between GQ distribution and semantic complexity; more precisely, the more complex a GQ, the less frequent it tends to be.

In order to test **(H)** we leverage on *multiple factor regression models* to reasonably quantify the predictive value of all such factors vis-à-vis quantifier frequency (cf. Gries, 2010).

Observation **(H)** is consistent with *the principle of least effort in communication*: speakers tend to minimize the communication effort by generating so-called “simple” messages. We take this result as an argument in favor of the claim, for instance defended by Szymanik (2016), that abstract semantic complexity measures may enrich the methodological toolbox of the language complexity debate.

We also considered whether semantic complexity can be clearly distinguished from syntactic or surface-form complexity. Clearly, semantics is not the only potential source of complexity in language (cf. Miestamo et al., 2008): surface-form length, syntax (e.g., nesting levels of subordinated clauses or parse tree depth), morphology (e.g., complex word inflection and derivation), monotonicity, and such, also all play a role (cf. Castello, 2008).

2. Semantic complexity of quantifiers

2.1. Quantifiers

What are the numerical expressions (generalized quantifiers, GQs) we are going to talk about? Intuitively, on the semantic level, quantifiers are expressions that appear to be descriptions of quantity, e.g., ‘all’, ‘not quite all’, ‘nearly all’, ‘an awful lot’, ‘a lot’, ‘a comfortable majority’, ‘most’, ‘many’, ‘more than k’, ‘less than k’, ‘quite a few’, ‘quite a lot’, ‘several’, ‘not a lot’, ‘not many’, ‘only a few’, ‘few’, ‘a few’, ‘hardly any’, ‘one’, ‘two’, ‘three’, etc. To concisely capture the semantics (meaning) of the quantifiers we should consider them in the sentential context, for instance:

(1) More than seven students are smart.
(2) Fewer than eight students received good marks.
(3) More than half of the students danced nude on the table.
(4) Fewer than half of the students saw a ghost.

[1] Thus, we ignore quantifiers as ‘not all’ or the distributive reading of ‘each’.

The formal semantics of natural language describes the meanings of those sentences. Sentences (1)–(4) share roughly the same linguistic form $\mathsf{Q}(A, B)$, where Q is a quantifier (a determiner), A is a predicate (the sentence's subject) denoting the set of students, and B is another predicate (the sentence's predicate) referring to various properties specified in the sentences. One way to capture the meanings of these sentences is by specifying their truth-conditions, saying what the world must be like in order to make sentences (1)–(4) true. To achieve this, one has to specify the relation introduced by the quantifier that must hold between predicates A and B. This is one of the main tasks of generalized quantifier theory (see, e.g., Peters and Westerståhl, 2006)—assigning uniform interpretations to the quantifier constructions across various sentences by treating the determiners as relations between sets of objects satisfying the predicates. We say that the sentence 'More than seven A are B' is true if and only if there are more than seven elements belonging to the intersection of A and B (card$(A \cap B) > 7$). Analogously, the statement 'Fewer than eight A are B' is true if and only if card$(A \cap B) < 8$. In the same way, the proposition 'More than half of the A are B' is true if and only if the number of elements satisfying both A and B is greater than the number of elements satisfying only A (i.e., card$(A \cap B) >$ card$(A - B)$) and then we can also formalize the meaning of sentence 'Fewer than half of the A are B' as card$(A \cap B) <$ card$(A - B)$.[2]

We are interested in the following: given a class of quantifiers (numerical concepts) realized in natural language can we categorize them with respect to their semantic complexity in an empirically plausible way?

2.2. Semantic complexity classes for quantifiers

The idea, proposed by van Benthem (1986), is to characterize the minimal computational devices that recognize different quantifiers in terms of the well-known Chomsky hierarchy. By recognition we mean deciding whether a simple quantifier sentence of the form $\mathsf{Q}(A, B)$ is true in a situation (model) M. Such devices define the *semantic complexity* of a quantifier Q. Let us explain what we mean with the models below:

2.2.1. Aristotelian quantifiers

Imagine that you have a picture presenting colorful dots and consider the following sentence:

(5) Every dot is red.

If you want to verify that sentence against the picture it suffices to check the color of all dots at the picture one by one. If we find a non-red one, then we know that the statement is false. Otherwise, if we analyze the whole picture without finding any non-red element, then the statement is true. We can easily compute the task using the following finite automaton from Fig. 1, which simply checks whether all elements are red. Such acyclic 2-state finite automata characterize the class of *Aristotelian* quantifiers.

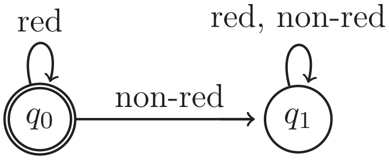


Fig. 1. Finite automaton for computing sentence (2.2). It inspects the picture dot by dot starting in the accepting state (double circled), $\boldsymbol{q_0}$. As long as it does not find a non-red dot it stays in the accepting state. If it finds such a dot, then it already 'knows' that the sentence is false and moves to the rejecting state, $\boldsymbol{q_1}$, where it stays no matter what dots come next. Obviously, as a processing model the automaton could terminate instantly after entering the state $\boldsymbol{q_1}$, however, we leave the loop on $\boldsymbol{q_2}$ following the convention of completely defining the transition function.

2.2.2. Counting quantifiers

In a very similar way, we can compute numerical quantifiers in the following sentences:

(6) More than three dots are red.
(7) Fewer than four dots are red.

If we want to verify the sentences against a picture, all we have to do is check the color of all the dots in the picture, one by one. If we find four red dots, then we know that statement (6) is true. Otherwise, if we analyzed the whole picture without

[2] Obviously, in many of these cases our truth-conditions capture only fragments of the quantifier meaning, or maybe we should better say, approximate typical meaning in natural language. For instance, we interpret 'most' and 'more than half' as semantically equivalent expression although there are clear differences in the linguistic usage. The point here is two-fold, on the one hand the same idea of generalized quantifiers can be used to capture various subtleties in the meaning, and even more importantly, from our perspective, majority of such extra-linguistic aspects, like pragmatic meaning, would not make a difference for the semantic complexity.

finding four red elements, then statement (7) is true. We can easily compute the task using the following finite automata from Figs. 2 and 3.[3] Finite state automata with $k+2$ states characterize the class of *counting* quantifiers.

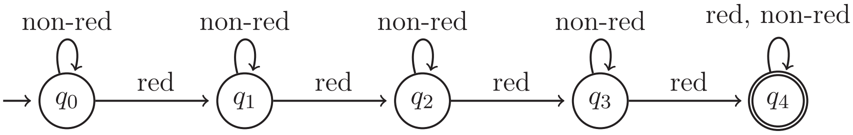


Fig. 2. This finite automaton decides whether more than three dots are red. The automaton needs five states. It starts in the rejecting state, $\boldsymbol{q_0}$, and eventually, if the condition is satisfied, moves to the double-circled accepting state, $\boldsymbol{q_4}$. Furthermore, notice that to recognize 'more than seven', we would need an analogous device with nine states.

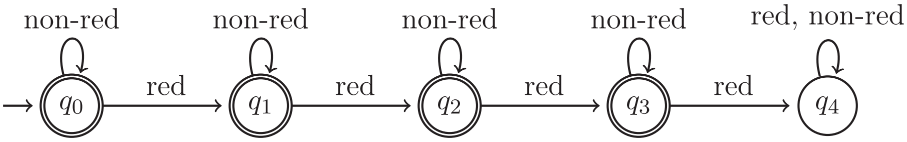


Fig. 3. This finite automaton recognizes whether fewer than four dots are red. The automaton needs five states. It starts in the accepting state, $\boldsymbol{q_0}$, and eventually, if the condition is not satisfied, moves to the rejecting state, $\boldsymbol{q_4}$. Furthermore, notice that to recognize 'fewer than eight', we would need an analogous device with nine states.

2.2.3. *Proportional quantifiers*

These finite automata introduced above are very simple and have only a very limited computational power. Indeed, they cannot recognize proportional quantifiers, which compare the cardinalities of two sets (van Benthem, 1986) as in the following sentences:

(8) More than half of the dots are red.
(9) Fewer than half of the dots are red.

As the pictures may contain any finite number of dots, it is impossible to verify those sentences using only a fixed finite number of states, as we are not able to predict beforehand how many states are needed. To give a computational device for this problem, an unbounded internal memory, which allows the automaton to compare two cardinalities, is needed. The device we can use is a push down automaton that 'counts' a number of red and non-red dots, stores them in its stack, and compares the relevant cardinalities (numbers) (see e.g., van Benthem, 1986).

Push-down automata can not only read the input and move to the next state, they also have access to the stack memory and depending on the top element of the stack they decide what to do next. Graphically, we represent it by the following labeling of each transition: $x, y/w$, where x is the current input the machine reads (i.e., the element under consideration), y is the top element of the stack, and w is the element which will be put on the top of the stack next (Hopcroft et al., 2000). For instance, the push-down automata from Fig. 4 computes sentence 'Fewer than half of the dots are red'. Furthermore, notice that to recognize 'more than half', we would need an almost identical device, the only difference being the reversed accepting condition: accept only if there is a red dot left on the top of the stack. Pushdown automata characterize the class of *proportional* quantifiers.

2.3. *Other semantic factors influencing quantifier difficulty*

The above described model characterizes quantifier meaning into two classes: regular quantifiers (recognizable by finite automata) and context-free quantifiers (recognizable by push-down automata). There is abundance of psycholinguistic evidence corroborating the significance of this distinction (see Szymanik, 2016, for an overview). However, clearly there are

[3] Formally speaking, the automata as input take strings encoding the finite situations (models). They are to decide whether a given quantifier sentence, $\mathsf{Q}(A,B)$, is true in the model. We restrict ourselves to finite models of the form $\mathbb{M} = (M, A, B)$. For instance, let us consider the model $\mathbb{M}$, where $M = \{c_1, c_2, c_3, c_4, c_5\}$, $A = \{c_2, c_3\}$, and $B = \{c_3, c_4, c_5\}$. As we are only interested in A elements we list c_2, c_3. Then we replace c_2 with 0 because it belongs to A but not B, and c_3 with 1 because it belongs to A and B. As a result, in our example, we get the word 10 that uniquely describes the model with respect to all the information needed for the quantifier verification in natural language. Now, we can feed this code into a finite automata corresponding to quantifiers. For instance, the automaton for 'Some A are B' will start in a rejecting state and stay there after reading 0. Next, it will read 1 and move to the accepting state. The PDA for the quantifier most, on the other hand, will compare the number of 0s and 1s and in this case end up in the rejecting state. Our encoding works under the implicit assumption that all quantifiers satisfy isomorphism, conservativity and extentionality that are strongly hypothesized to be among quantifier semantic *universale* (Barwise and Cooper, 1981; Peters and Westerståhl, 2006). For quantifiers do not satisfying these properties we would need to take into account all elements belonging to the model (see Mostowski, 1998).

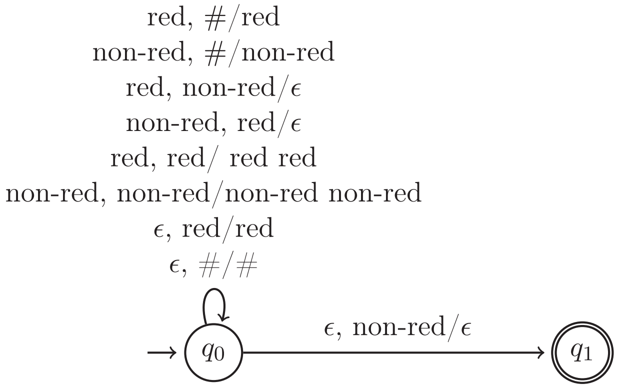


Fig. 4. This push-down automaton recognizes whether fewer than half of dots are red. The automaton needs two states and the stack. It starts in the accepting state, $\boldsymbol{q_0}$ with an empty stack marked by #. If it finds a red dot it pushes it on top of the stack and stays in $\boldsymbol{q_0}$, if it finds a non-red dot it also pushes it on top of the stack. If it finds a red (non-red) dot and there is already non-red (red) dot on the top of the stack, the automaton pops out the top of the stack (by turning it into the empty string ε), i.e., it 'cancels' dot pairs of different colors. If it sees a red (non-red) dot and there already is a dot of the same color on the stack, then the automaton pushes another dot of that color on the top of the stack. Eventually, when the automaton has analyzed all the dots (input $= \varepsilon$) then it looks on the top of the stack. If there is a non-red dot it moves to the accepting state, otherwise it stays in the rejecting state.

other factors and sources of linguistic complexity besides semantic complexity that contribute to the cognitive difficulty of language processing, and that may be reflected in the distribution of particular lexical items, e.g., length in the number of characters or words. Furthermore, semantic complexity provides a relatively coarse-grained grouping of quantifiers that ignores or abstracts away other semantic properties. In this section let us discuss two important semantic properties of quantifiers that are widely believed to be implicated in the complexity of quantifier processing.

2.3.1. *Monotonicity*

We will say that a quantifier Q is upward monotone (increasing) in its left (respectively, right) argument if and only if, for any sets A, B and B', if B is a subset of B' (i.e., $B \subseteq B'$), then $\mathsf{Q}(B,A)$ entails $\mathsf{Q}(B',A)$ (respectively, $\mathsf{Q}(A,B)$ entails $\mathsf{Q}(A,B')$). For example, the quantifier 'some' is upward monotone in its left argument as sentence (10) implies sentence (11) and the quantifier 'most' is monotone increasing in its right arguments as sentence (12) implies sentence (13), and the sets of boys and very happy students (B) are subsets of the sets of children and happy students (B'), respectively.

(10) Some boys are happy.
(11) Some children are happy.
(12) Most students are very happy.
(13) Most students are happy.

Note that some quantifiers are neither upward nor downward monotone, in any argument. For example, consider the following sentences:

(14) Exactly five boys are happy.
(15) Exactly five boys are very happy.
(16) Exactly five children are happy.

Neither does sentence (14) entail sentence (15) nor vice versa, and the same is true about sentence (14) and (16). We will say that *a quantifier* Q *is monotone* if it is upward or downward monotone in one of its arguments. Otherwise, we will call it non-monotone.

Monotonicity is widely believed to be a key property of natural language (see, e.g., Peters and Westerståhl, 2006). Especially, there is an ample experimental evidence that downward monotone quantifiers are harder to process than upward monotone ones (see, e.g., Moxey and Sanford, 1993; Geurts and van der Silk, 2005; Szymanik and Zajenkowski, 2013).

2.3.2. *Comparative/superlative distinction*

Our complexity measure identifies as equally complex expressions with the same meaning. As we already mentioned, sometimes such simplification may seem problematic. Consider the comparative quantifier 'more than 3' and the equivalent superlative quantifier 'at least 4'. From the semantic automata point of view they are both associated with the same minimal computational device and by extension have the same complexity index. However, there are independent reasons to think that superlative quantifiers have richer meanings than comparative ones (see, e.g., Geurts and Nouwen, 2007). Indeed, there is experimental evidence suggesting that superlative quantifiers are harder to process, give rise to different patterns of reasoning, and are acquired later by children, than the corresponding comparative quantifiers (see Geurts et al., 2010; Cummins and Katsos, 2010).

3. Corpus analysis

3.1. Methods

In this section we outline the analysis of English generalized quantifiers' frequency in corpora, see Table 1. We do it by identifying quantifier surface forms in a very large English corpus. From an implementation side of things, we relied on (buffered) server side scripts to explore our large corpus and collect frequency statistics.[4] The frequency statistics collected were subsequently used for regression analysis carried out using the R statistical package.[5] Given that frequency is a count variable we used specific regression models, generalized linear models, to carry out an analysis of residual deviance to quantify the influence of each factor (cf. Dobson and Barnett, 2008).

3.2. Corpus

In order to obtain a representative sample we considered a very large English corpus covering multiple domains and sentence types (declarative and interrogative), the WaCkypedia_EN (WaCky) corpus, built and curated by Baroni et al. (2009). The WaCky corpus was built by postprocessing a full 2009 dump of Wikipedia, and is freely available for download and use.[6] The authors removed all HTML markup and image files, and filtered out web-pages devoid of real textual content (e.g., tables displaying statistics), until a balanced (relatively to subject matter or domain, vocabulary, sentence type and structure, etc.) and representative corpus of English was achieved, see Fig. 5. For full details, please refer to the paper by Baroni et al. (2009).

Baroni et al. segmented, tokenized, part-of-speech annotated and parsed the corpus using the TreeTagger[7] (part-of-speech annotation) and MaltParser[8] (parser) statistical systems, that have an accuracy of over 90%, resulting in a dataset that exhibit the format shown in Fig. 5. For each sentence, the corpus provides the following information: **(a)** the list of its tokens (first column), **(b)** the list of their corresponding lemmas or morphological stems (second column), **(c)** the list of their corresponding part-of-speech (POS) tags (third column), **(d)** information regarding the position of the words in the sentence (fourth and fifth columns), and **(e)** the list of their corresponding typed (syntactic) dependencies (fifth column). For our experiments, we took into consideration only **(a)**–**(c)**. The POS tags used by TreeTagger are derived from the well-known Penn Treebank list of POS tags.[9]

```
<s>
Flender Flender NP      1       3       VMOD
Werke   Werke   NP      2       3       SBJ
was     be      VBD     3       0       ROOT
a       a       DT      4       7       NMOD
German  German  JJ      5       7       NMOD
shipbuilding    shipbuilding    NN      6       7       NMOD
company company NN      7       3       PRD
,       ,       ,       8       7       P
located locate  VVN     9       7       NMOD
in      in      IN      10      9       ADV
Lübeck  Lübeck  NP      11      10      PMOD
.       .       SENT    12      0       ROOT
</s>
```

Sentences	∼ 43 million
Tokens	∼ 800 million

Fig. 5. Left: Sample tokenized, POS-annotated and dependency-parsed sentence from the WaCky corpus. Note the use of Penn Treebank POS tagset. Annotations are tab-separated. Right: WaCky corpus statistics.

3.3. Patterns and features

3.3.1. Patterns

We identified generalized quantifiers indirectly, via multiword part-of-speech (POS) tagged *patterns* (regular expressions) that approximate their surface forms. A POS tag is a label that describes the syntactic role of a word within a sentence. This information is key to disambiguate such surface forms, which are usually polysemous and do not always denote a generalized quantifier. For instance, the word 'most' in the WaCky corpus can occur as determiner (tag DT) or a (comparative or superlative) adverb (tags RBR and RBS, resp.), and need not always denote the (proportional) quantifier *most*. For instance, it denotes a quantifier when followed by a plural noun (tag NNS) as in 'most/DT men/NNS', but not when followed by an

[4] https://github.com/camilothorne/my-wacky-analysis.

[5] https://www.r-project.org/.

[6] http://wacky.sslmit.unibo.it/doku.php.

[7] http://www.cis.uni-muenchen.de/schmid/tools/TreeTagger/.

[8] http://www.maltparser.org/.

[9] See: http://www.ling.upenn.edu/courses/Fall_2007/ling001/penn_treebank_pos.html.

adjective (tag JJ) as in ‘most/RBS grateful/JJ’. Also, while some GQs are expressed by a single word or unigram (e.g., *some* is expressed by the determiner ‘some’—an unigram), others require more than one token. In particular counting and proportional quantifiers (e.g., $>k$ can be expressed by ‘more than k’—a multiword expression consisting of at least 3 words) can be considered multiword or n-gram quantifiers.

For each of the GQs considered in our study, we defined one or more such patterns, *disjoint* from all others. The selection criteria for the patterns was their linguistic plausibility. We *lower-cased* all the input sentences (words and POS tags) and focused on lemmas whenever possible to avoid multiplying patterns due to inflection. For instance, for the GQ $> k$ we matched over the corpus the following patterns or regular expressions[10]:

- ‘at/in least/jjs [a-z]{1,12}/cd’, viz., the preposition ‘at’ followed by the superlative adjective ‘least’ and a cardinal comprising up to 12 characters;
- ‘more/rbr than/in [a-z]{1,12}/cd’, viz., the comparative adverb ‘more’ followed by the preposition ‘than’ and a cardinal;
- ‘more/jjr than/in [a-z]{1,12}/cd’, viz., the same as before, but with ‘more’ a comparative adjective.

This gave rise to a total of 36 GQ patterns, that can be seen in Table 2. We subsequently counted the number of times the pattern is instantiated within a sentence in the corpus, that is, its number of *occurrences*.

Table 2
GQ patterns and features for regression analysis.

	GQ pattern	Frequency	Rank	GQ	Class	Type	Left mon.	Right mon.	Length (words)	Length (chars.)
1	every/dt	246492	5	all	ari	NA	down	up	1	5
2	all/dt	710149	2	all	ari	NA	down	up	1	3
3	all/pdt	168661	8	all	ari	NA	down	up	1	3
4	each/dt [a-z]{1,12}/nn	200337	7	all	ari	NA	down	up	1	4
5	no/dt	464755	4	all	ari	NA	down	down	1	2
6	some/dt	742134	1	some	ari	NA	up	up	1	4
7	more/rbr than/in [a-z]{1,12}/cd	81	20	>k	cnt	comp	up	up	3	8
8	more/jjr than/in [a-z]{1,12}/cd	24826	10	>k	cnt	comp	up	up	3	8
9	at/in least/jjs [a-z]{1,12}/cd	26210	9	>k	cnt	supe	up	up	3	7
10	less/rbr than/in [a-z]{1,12}/cd	0	31	<k	cnt	comp	down	down	3	8
11	fewer/jjr than/in [a-z]{1,12}/cd	0	31	<k	cnt	comp	down	down	3	9
12	at/in most/jjs [a-z]{1,12}/cd	609	17	<k	cnt	supe	down	down	3	6
13	less/jjr than/in [a-z]{1,12}/cd	5344	12	<k	cnt	comp	down	down	3	8
14	fewer/jjr than/in [a-z]{1,12}/cd	0	31	<k	cnt	comp	down	down	3	9
15	most/dt	0	31	most	pro	comp	none	up	1	4
16	more/rbr than/in half/nn	121	18	>1/2	pro	comp	none	up	3	12
17	most/jjs [a-z]{1,12}/nns	675723	3	most	pro	comp	none	up	1	4
18	most/rbs [a-z]{1,12}/nns	11699	11	most	pro	comp	none	up	1	4
19	more/jjr than/in half/nn	928	14	<1/2	pro	comp	none	up	3	12
21	less/rbr than/in half/nn	27	21	<1/2	pro	comp	none	down	3	12
22	fewer/jjr than/in half/nn	0	31	<1/2	pro	comp	none	down	3	13
23	few/jj [a-z]{1,12}/nns	209324	6	few	NA	NA	NA	NA	1	3
24	at/in least/jj half/nn	0	31	>1/2	pro	supe	none	up	3	11
25	more/rbr than/in half/nn	121	18	>1/2	pro	comp	none	up	3	12
26	more/jjr than/in half/nn	928	14	>1/2	pro	comp	none	up	3	12
27	at/in least/jjs [a-z]{1,12}/cd [a-z]{1,12}/nns of/in	969	13	>p/k	pro	supe	none	up	5	9
28	more/rbr than/in [a-z]{1,12}/cd [a-z]{1,12}/nns of/in	3	25	>p/k	pro	comp	none	up	5	12
29	more/jjr than/in [a-z]{1,12}/cd [a-z]{1,12}/nns of/in	636	16	>p/k	pro	comp	none	up	5	10
30	less/rbr than/in half/nn	27	21	<1/2	pro	comp	none	down	3	12
31	fewer/jjr than/in half/nn	0	31	<1/2	pro	comp	none	down	3	13
32	at/in most/jjs half/nn	6	24	<1/2	pro	supe	none	down	3	10
33	less/rbr than/in [a-z]{1,12}/cd [a-z]{1,12}/nns of/in	0	31	<p/k	pro	comp	none	down	5	10
34	fewer/jjr than/in [a-z]{1,12}/cd [a-z]{1,12}/nns of/in	0	31	<p/k	pro	comp	none	down	5	11
35	at/in most/jjs [a-z]{1,12}/cd [a-z]{1,12}/nns of/in	10	23	<p/k	pro	comp	none	down	5	8
36	at/in most/rbs [a-z]{1,12}/cd [a-z]{1,12}/nns of/in	0	31	<p/k	pro	comp	none	down	5	8

3.3.2. Features

To understand to what extent semantic complexity influences GQ distribution in the WaCkY corpus, but also, how it interacts with other potential sources of complexity—in particular, surface-form complexity—we observed, for each pattern, the following features.

[10] The reader can find a detailed explanation of regular expression syntax in http://web.mit.edu/hackl/www/lab/turkshop/slides/regex-cheatsheet.pdf.

(1) *GQ (pattern) frequency*: a discrete numeric feature, the frequency counts for each pattern.
(2) *GQ (pattern) frequency rank*: a discrete numeric feature, used to sort GQ patterns by frequency order.
(3) *GQ (pattern) class*: an ordered factor encoding GQ class, with three values: 'ari' (Aristotelian), 'cnt' (counting), and 'pro' (proportional);
(4) *GQ (pattern) right monotonicity*: a factor encoding the monotonicity properties of the right argument (noun or noun phrase) of the GQ, with three values: 'up' (upward monotonic), 'down' (downward monotonic), and 'none' (non-monotonic).
(5) *GQ (pattern) left monotonicity*: a factor encoding the monotonicity properties of the left argument (noun or noun phrase) of the GQ.
(6) *GQ (pattern) type*: a Boolean factor that encodes whether the GQ is comparative ('comp') or superlative ('supe').
(7) *GQ (pattern) length in words*: an ordered factor, that clusters GQ patterns according to the *minimum* number of word tokens in the *unlabeled* (multiword) expression underlying the pattern.[11]
(8) *GQ (pattern) length in characters*: a discrete numeric feature that provides an estimate of the *minimum* number of characters of the *unlabeled* (multiword) expression underlying the GQ pattern.[12]

Table 2 summarizes the feature values observed for each of the 36 GQ patterns studied. While many more potential features (e.g., level of nesting in subordinated phrases, position in sentence, type of noun phrase, etc.) are conceivable, we believe that the above 8 are sufficient for making the conceptual point we are aiming at.

3.4. Descriptive analysis

3.4.1. Distributions

The distributions observed are summarized by Figs. 6 and 7. Aristotelian quantifiers occur more frequently than counting quantifiers, and counting quantifiers than proportional quantifiers. It can be pointed out however that a similar bias seems to

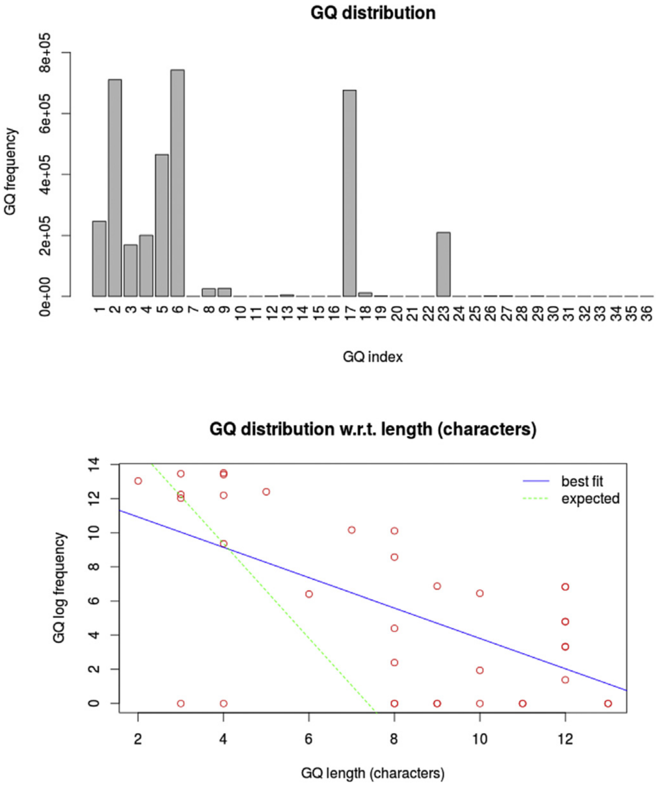
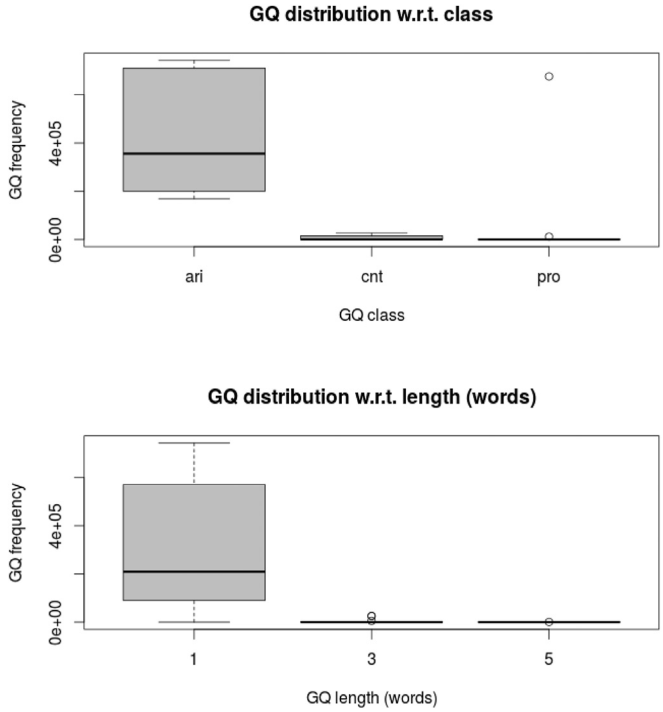


Fig. 6. Top: frequency distribution of the GQ patterns. The indexes refer to the GQ pattern indexes/IDs in Table 2. When ordered by frequency rank, we can observe a non-normal, right-tailed distribution. Bottom: GQ distribution by character length. The green line describes the expected geometric decrease. Notice that GQ frequency (log-scale) decreases slower than expected.

[11] E.g., 'few', derived from the pattern 'few/jj [a-z]{1,12}/nns', comprises 1 token.

[12] E.g., 'every', of 5 characters, derived from the pattern 'every/dt'.


GQ distribution w.r.t. class

GQ distribution w.r.t. length (words)

Fig. 7. GQs by class and length (boxplots).

exist towards short (unigram or single token) quantifiers, and multiword quantifiers are very rare. Hence, both GQ length and semantics seem to influence GQ distribution in our dataset.

3.4.2. *Length and surface-level complexity*

The boxplots in Fig. 7, show a similarly-shaped frequency distribution of quantifiers when aggregated either by class or by length. Does this suggests that GQ distribution may be explained in terms of superficial features such as surface-form length? Both Miller et al. (1958) and Piantadosi (2015) argue that word distribution decreases geometrically w.r.t. word length. Applied to quantifiers, this would imply that GQ frequency of any quantifier $fr(\mathtt{Q})$ can be described by the equation:

$$fr(\mathtt{Q}) = \left|W\right| \cdot p^{char(\mathtt{Q})} \cdot (1-p) \tag{1}$$

where $p \approx \frac{1}{27}$ (the probability of hitting a given character) and $char(\mathtt{Q})$ denotes the length of GQ pattern $\mathtt{Q}$ in number of characters.

However, as Fig. 6 (right plot) shows, while GQ pattern frequency seems to diminish w.r.t. length (measured in number of characters), when compared to the frequency predicted by Equation (1), such decrease is slower than expected. Additionally, one can observe (red circles) large frequency variability for most character lengths. Furthermore, while expressions such as ‘at most one quarter of’ (of length 5) are less common than expressions like ‘most’ (of length 1), when looking at each length group (length 1, length 2, length 5), Table 2 suggests that lower complexity GQs outnumber higher complexity GQs. This suggests a distribution that is not influenced by solely the surperficial complexity—the length of its surface form—of a GQ expression, but by an array of diverse features whose impact can be best understood via regression analysis.

3.5. *Regression analysis*

3.5.1. *Generalized linear models*

Classical regression models assume that the predicted variable Y in a regression—GQ frequency in our case—is normally distributed, viz., $Y \sim \mathcal{N}(\mu, \sigma)$. As Fig. 6 shows it is not the case for our GQ dataset. This does not preclude the existence of a dependency between GQ frequency and the features or factors described in Section 3.3. However, in order to discover such potential dependencies a more powerful regression models should be used. These models assume different distributions for Y and posit that Y is not the direct result of a linear combination of factors or features (as in standard linear regression models), but rather a more complex function. In what follows, we use a class of regression models particularly suited to the analysis of count-frequency data (cf. Dobson and Barnett, 2008).

Definition 1 (Generalized linear model (GLM)).

A (multifactor, mixed) generalized linear model (GLM) has the form

$$f(Y) = \theta_1 X_1 + \cdots + \theta_k X_k + \theta_{k+1} \tag{2}$$

where

(i) $f : \mathbb{R} \to \mathbb{R}$ *is a* link *function,*
(ii) $Y \sim \mathscr{D}$, *with* $\mathscr{D}$ *an arbitrary distribution.* ‡

Y is the *response* variable, $X_1, \ldots, X_n$ its *predictors* (features), and $\theta_1, \ldots, \theta_k, \theta_{k+1}$ are the *parameters* (coefficients) of the model, The X_i s must be conditionally independent w.r.t. Y, viz., $(Y|X_i) \perp (Y|X_{i+1})$ for the model to applicable. The link function—typically, the natural logarithm ln—is used to model Y as the result of a *transformation* of $\theta_1 X_1 + \cdots + \theta_k X_k + \theta_{k+1}$. Thus, a GLM models dependencies that, while not *per se* linear, become linear *modulo* the link function.[13] Random effects (cf. McCulloch, 1997) stand for noisy features that, while still holding some predictive value, are allowed to vary widely with each observation. There are two main families of GLMs for count data (cf. Dobson and Barnett, 2008):

- Poisson: Assumes that $Y \sim \mathscr{P}(\lambda)$, where $\mathscr{P}(\lambda)$ is a Poisson distribution, and that frequency is heavily right-tailed (skewed to the left).
- Negative binomial: Assumes that $Y \sim \mathscr{NB}(r,p)$, where $\mathscr{NB}(r,p)$ is a negative binomial distribution, and that frequency decreases geometrically.

3.5.2. *Feature selection*

To ensure conditional independence, predictor variables in GLMs must be *independently distributed*. Several techniques exist to test for independence. We resort to (normalized) mutual information—a measure derived from information theory that quantifies how similar is the distribution of two categorical, ordinal, or discrete variables X and Y, as the ones observed (features 3–8) in this paper. This measure is also suitable for corpus statistics (cf. Gries, 2010). The MI score defines a similarity metric over X and Y, ranging from 0 (independently distributed) to 1 (identically distributed), with the usual properties of similarity metrics.

Definition 2 (Mutual information (MI)).

The normalized mutual information (MI) of two discrete random variables X and Y is defined by

$$I_n(X;Y) = \frac{H(X) - H(X|Y)}{H(X,Y)} \tag{3}$$

where $H(X)$, $H(X|Y)$ *and* $H(X,Y)$ *are the usual notions of the Shannon entropy, conditional entropy, and joint entropy of X and Y of (Shannon) information theory (cf.* Manning and Schütze, 2000*).* ‡

Table 3 describes the MI scores for each pair of the 8 features described in Section 3.3 and observed for each GQ. There is a high (≥ 0.5) normalized MI between GQ class and, on the one hand, the number of characters (character length) of GQs, and, on the other hand, the monotonicity properties of their left arguments. Additionally, there is a very high normalized MI between GQ length measured in terms of their number of tokens and in terms of characters (0.70), but the same does not hold w.r.t. GQ class (0.30). This suggests retaining as frequency predictors for our GLMs only *four* features, namely: **(a)** GQ class, **(b)** GQ type, **(c)** GQ length (in number of tokens) and **(d)** GQ right monotonicity.

Table 3
Normalized MI table for the features described in Table 2 (excluding rank). In bold, the scores > **0.5** (violating conditional independence).

	Class	Left mon.	Right mon.	Len. (words)	Len. (chars.)	Type
Class		**0.56**	0.02	0.30	**0.58**	0.24
Left mon.			0.05	0.10	0.42	0.09
Right mon.				0.10	0.31	0
Len. (words)					**0.70**	0.10
Len. (chars.)						0.14

3.5.3. *Model comparison*

We fitted Poisson and negative binomial GLMs with GQ frequency as response variable and GQ class, type, length (in number of tokens), and GQ right monotonicity as predictors. Moreover, we tested if length was a random effect, by considering for each model a mixed counterpart. More precisely:

[13] Since $\ln(Y) = \theta_1 X_1 + \ldots + \theta_k \Leftrightarrow Y = \exp(\theta_1 X_1) \cdots \exp(\theta_k)$.

(1) a Poisson model with *length* as random effect and the other features as fixed effects (POISSON-MIXED);
(2) a Poisson model where all the features are fixed effects (POISSON).
(3) a negative binomial model with *length* as random effect and the other features as fixed effects (BINOM-MIXED);
(4) a negative binomial model where all the features are fixed effects (BINOM).

We considered POISSON-MIXED as a baseline model. For the negative binomial models, we considered models with$r = 1$.[14] The idea behind considering length as a random feature is that: if GQ frequency were explained by semantic complexity alone, there would be no substantial difference between the fixed model and its respective mixed counterpart.

Table 4 (right) summarizes the results obtained. Two quantities were used to define the ranking:

(a) We measured the Akaike information criterion (AIC) of the models, that describes the goodness of fit of (multinomial) regression models. The lower this number (and closer to zero), the better the fit (cf. Dobson and Barnett, 2008). **(b)** We measured the (statistical) significance level for each AIC number. All models showed a (very strongly) significant decrease in AIC w.r.t. to POISSON-MIXED, with BINOM showing the best scores.

Table 4
Left: AOD table for the best model (BINOM). GQ class, GQ length, and right monotonicity have a statistically (very strongly) significant impact on GQ frequency. GQ length has the highest impact, followed by class, and then monotonicity. Right: Comparison of regression models. While all models significantly improve on the baseline (Poisson mixed model), the negative binomial model with *fixed* effects shows the best AIC fit score (highlighted in gray).

Feature	Deviance	p-Value
Length (words)	47.06%	$3.47\cdot e^{-10}$
Class	27.29%	$5.25\cdot e^{-7}$
Type	0.02%	0.97
Right mon.	25.65%	$1.15\cdot e^{-6}$

Model	AIC	p-Value
POISSON-MIXED	1446287.3	(Baseline)
POISSON	1446000.0	$1.191\cdot e^{-10}$
BINOM-MIXED	426.7	$<2.2\cdot e^{-16}$
BINOM	409.3	$<2.2\cdot e^{-16}$

3.5.4. Analysis of deviance (AOD)

As GLMs do not assume data to be normally distributed it is not possible to apply a standard analysis of variance to understand the impact of the factors on GQ frequency (cf. Dobson and Barnett, 2008). Therefore, rather than analyzing the *variance* of the residuals, we carry out an *analysis of deviance* (AOD), in which we look at GQ frequency deviance w.r.t. the model's predicted regression line, and seek for each independent factor in the model to **(a)** test for statistically significant influence **(b)** quantify its impact on deviance.

Table 4 (left) summarizes the results yielded by BINOM. As expected both superficial features (length) and semantic features had an (approximately equal) impact. Length (in number of words) explains by itself 47.06% of the deviance, followed by GQ class (27.29%) and (right) monotonicity (25.65%)—the latter accounting together for more than 52% of deviance. In all three cases, the influence is also statistically significant.

3.6. Discussion

3.6.1. Contributions

We have analyzed the distribution of GQs in the WaCky corpus using regression analysis and quantifier patterns. Our results agree with the prediction **(H)**. The frequency observed does not describe a normal distribution but rather a right-tailed distribution that is skewed towards both Aristotelian (i.e., low semantic complexity) and unigram (low string complexity) quantifiers. Furthermore, by observing an array of semantic (GQ class, type, monotonicity) and surface-level factors (GQ length) we have been able to fit a negative binomial GLM wherein semantic factors (GQ class, monotonicity) explain 52.92% of the frequency variability, of which 27.29% by semantic complexity alone. Interestingly, monotonicity has an impact similar to semantic complexity ($25,65\%$), while the superlative/comparative distinction has none. Our model shows also that length is an important factor, explaining the remaining 47.06% frequency variability. However, length alone cannot explain the distribution. Indeed, frequency decreases (or 'decays') much slower than a theoretical model based only on string length would predict. These results are statistically significant. This last fact seems to be further substantiated by Table 3, that indicates length and semantic complexity have a high (0.58) MI—indicating a link between these two factors. Altogether, this seems to indicate that semantic complexity is an important aspect underlying GQ distributions as observed

[14] Setting the negative binomial distribution parameter r to 1 improves fits for datasets where some events or outcomes are very frequent, but most are rare (cf. Dobson and Barnett, 2008), as in our GQ dataset.

in (large) corpora, and in particular when combined with monotonicity to produce more fine-grained semantic complexity distinctions.

These results are in general consistent with the so-called *principle of least effort* in human communication: Speakers are endowed with only finite (and limited) linguistic resources which drive them to continuously seek to minimize their effort to generate a message by using few, short, ambiguous words and short sentences (see Miller et al., 1958; Piantadosi, 2015). This empirical principle, first formulated by G. K. Zipf in the 1940s, is believed to explain why the (frequency) distribution of linguistic expressions and constructs in natural language datasets is usually skewed towards 'simple' expressions and constructs, giving rise in some cases to power laws (i.e., distributions where frequency decreases geometrically w.r.t. frequency rank). Our results suggest that the notion of 'simplicity' assumed by the principle of least effort can be enlarged to take into account not only the length of quantifier expressions, but also their meaning, and specifically their formal semantics and the automata-based semantic complexity models to which they can be associated.

We also note that regression models are predictive models that, in particular, imply an asymmetric relationship between dependents variables (GQ frequency) and independent factors (GQ class, type, monotonicity and length). While this does not imply a causal link, it suggests a relation stronger than that of a simple correlation, and by extension some degree of influence of semantic complexity over the GQ distribution observed in the WaCky corpus.

3.6.2. Limitations of this study

3.6.2.1. Complexity measure. Our semantic complexity measure is defined in terms of the minimal computational device that can recognize a quantifier. This computational problem resembles sentence-picture verification and was successfully used to model such psycholinguistics tasks (see, e.g., Szymanik, 2016). However, one may ask how verification complexity relates to language production as reflected by corpus data. We admit that the relation may be indirect at best. We believe however that differences in semantic complexity capture a sort of intrinsic combinatorial complexity of quantifier meanings that should be to some extent reflected in linguistic tasks such as reasoning, comprehension, or production, etc. (see Isaac et al., 2014; Feldman, 2000, for more discussion).

Moreover, the semantic complexity measure abstracts away from the linguistic realization of meaning. For instance, the numeral quantifiers: 'some' and '(at least) one' are both recognized by the same two-state finite automaton (cf. discussion of comparative and superlative quantifiers). Hence, from the perspective of semantic complexity, the classes of Aristotelian and numerical quantifiers do overlap (although in a minimal way). But this is not a problem for our analysis as we are mainly interested in the difference between regular quantifiers (Aristotelian and numerical) and context-free quantifiers (proportional).

Related to the above point is the observation that numerical quantifiers do not form a uniform class because the number of states of 'exactly/less than/more than *n*' depends on *n*. This is again not that problematic for our analysis as we are not making the assumption that the number of states in the minimal automaton should be directly reflected in frequencies. That is, we are not claiming that 'at least thousand' is thousand times more complicated than 'at least one'. We are rather interested in the distinction between regular and context-free quantifiers.

These examples can be used to make an additional point: 'at least 1000' is far more frequent than 'at least 999' although it corresponds to the automaton with only one more state. Obviously, this is for pragmatic reasons, and it is a matter of future research to include pragmatic factors within our analysis.

Finally, from the semantic complexity point of view negation makes no difference, i.e., if $\mathtt{Q}$ is regular (context-free) quantifier, then also $\neg\mathtt{Q}$ and $\mathtt{Q}\neg$ are regular (context free). This is why we ignore negation in our corpora search. Note, however, that we take into account a related factor of monotonicity.

3.6.2.2. Corpus study. We would like to stress that our corpus study sacrifices coverage at the expense of precision: we study only a handful of GQs. In particular, we focused on a small subset of the GQs discussed by Barwise and Cooper (1981) and Szymanik (2016) that roughly coincides with those expressible in everyday or colloquial English. They are relatively easy to identify and appear often in generic English corpora such as WaCky.

Furthermore, we also point out the usual limitations of corpus studies that rely on very large annotated corpora. Firstly, annotation quality is limited: it is not feasible to annotate completely by hand very large corpora, but rather via machine learning techniques which are not very accurate. Secondly, annotation practice may diverge from linguistic theory: some theoretically accepted GQ surface forms, e.g., 'most/DT' (i.e., 'most' as a determiner, proposed by Barwise and Cooper (1981)), do not occur in the WaCky corpus at all, as they were never annotated as such (either by the human annotators or the automated annotators). Thus, the distribution described in Fig. 6 should be taken into consideration *cum grano salis*. Using higher quality though smaller corpora (e.g., the Brown corpus or the British National Corpus) might help to overcome these limitations, but this time at the expense of coverage,[15] yielding less meaningful statistics.

Last but not least, GLMs provide only approximate fits to the real distribution of the data, in particular when inferred over a relatively small number of observations (viz., 36). Thus, the results described in Table 4 should be taken with some caution.

[15] The British National Corpus contains only 100 million words, as compared to the more than 800 of WaCky, and is not freely accessible either.

4. Conclusions

Our results show that semantic complexity is associated with quantifier distribution in large English encyclopedic corpora such as the WaCky corpus. Negative binomial (generalized) regression models indicate that 27.09% of quantifier frequency variability in large textual datasets can be explained by quantifier complexity (and up to 52.14% by semantic factors, complexity included). The usefulness of computational approaches to assess the intricate complexity of linguistic expressions gathers additional support from experimental studies in psycholinguistics (see, e.g., Szymanik, 2016), and corpora analysis driven by deep semantic parsing in (Thorne, 2012).

The results also contribute to the discussion of semantic universals for natural language quantifiers (see Barwise and Cooper, 1981; Peters and Westerståhl, 2006). It seems that the answer to the question of which logically possible quantifiers are realized (and how often) in natural language depends not only on some formal properties of quantifiers, like monotonicity, but also on the computational complexity of the underlying semantic concepts. In other words, some quantifiers may not be realized in natural language (or be used very rarely) due to their semantic complexity (see also Kontinen and Szymanik, 2008; Szymanik, 2010).

As we mentioned in the introduction, our goal was to give a proof-of-concept of the applicability of abstract computational complexity measures to quantify semantic complexity. In the future, we would like to refine the results shown here by considering a much larger set of quantifiers, by *discovering and counting all* the potential quantifiers that occur in a corpus (while observing the features considered in this paper), to recognize stronger statistical patterns. Such goal can be achieved by, e.g., analyzing the parse tree of each sentence of the WaCky corpus to individuate the subtree that corresponds to the $\mathtt{Q}$ in $\mathtt{Q}(A, B)$. Additionally, we would like to use semantic complexity in the discussion of the *equivalent complexity thesis*: all natural languages are equally complex or have equal descriptive power (see, e.g., Miestamo et al., 2008). We believe we can contribute to this debate by performing analysis similar to these described here but over *multilingual* corpora and *creole/full language* pairs in order to check if all languages realize equally complex (e.g., context-free) semantic constructions such as, e.g., proportional quantifiers, and if they have similar distributions (realize equally complex expressions equally often).

Acknowledgments

The authors thank the anonymous reviewers for their comments and remarks to earlier versions of this paper. The research of the first author leading to these results has received funding from the European Research Council under the European Union's Seventh Framework Programme (FP/2007–2013)/ERC Grant Agreement n. STG 716230 CoSaQ.

References

Baroni, M., Bernardini, S., Ferraresi, A., Zanchetta, E., 2009. The WaCky wide web: a collection of very large linguistically processed web-crawled corpora. Lang. Resour. Eval. 43 (3), 209–226.
Barwise, J., Cooper, R., 1981. Generalized quantifiers and natural language. Linguist. Philos. 4, 159–219.
van Benthem, J., 1986. Essays in Logical Semantics. Reidel.
Castello, E., 2008. A corpus-based study of text complexity. In: Torsello, C.T., Ackerley, K., Castello, E. (Eds.), Corpora for University Language Teachers. Peter Lang, pp. 183–198.
Cummins, C., Katsos, N., 2010. Comparative and superlative quantifiers: pragmatic effects of comparison type. J. Semant. 27 (3), 271–305.
Dobson, A.J., Barnett, A.G., 2008. An Introduction to Generalized Linear Models. CRC Press.
Everett, D., 2005. Cultural constraints on grammar and cognition in Pirahã. Curr. Anthropol. 46 (4), 621–646.
Feldman, J., 2000. Minimization of boolean complexity in human concept learning. Nature 407 (6804), 630–633.
Geurts, B., Katsos, N., Cummins, C., Moons, J., Noordman, L., 2010. Scalar quantifiers: logic, acquisition, and processing. Lang. Cogn. Processes 25 (1), 244–253.
Geurts, B., van der Silk, F., 2005. Monotonicity and processing load. J. Semant. 22, 97–117.
Geurts, B., Nouwen, R., 2007. 'At least' et al.: the semantics of scalar modifiers. Language 83 (3), 533–559.
Gries, S.T., 2010. Useful statistics for corpus linguistics. In: Sánchez, A., Almela, M. (Eds.), A Mosaic of Corpus Linguistics: Selected Approaches. Peter Lang, pp. 269–291.
Hopcroft, J.E., Motwani, R., Ullman, J.D., 2000. Introduction to Automata Theory, Languages, and Computation, second ed. Addison Wesley.
Isaac, A., Szymanik, J., Verbrugge, R., 2014. Logic and complexity in cognitive science. In: Baltag, A., Smets, S. (Eds.), Johan van Benthem on Logic and Information Dynamics, Volume 5 of Outstanding Contributions to Logic. Springer International Publishing, pp. 787–824.
Juola, P., 1998. Measuring linguistic complexity: the morphological tier. J. Quantitative Linguist. 5 (3), 206–213.
Keenan, E.L., Paperno, D., 2012. Handbook of Quantifiers in Natural Language, vol. 90. Springer.
Kontinen, J., Szymanik, J., 2008. A remark on collective quantification. J. Log. Lang. Inf. 17 (2), 131–140.
Manning, C., Schütze, H., 2000. Foundations of Statistical Natural Language Processing. The MIT Press.
McCulloch, C.E., 1997. An Introduction to Generalized Linear Mixed Models. Technical Report, Report Num. BU-1340-MA. Biometrics Unit and Statistics Center, Cornell University.
McWhorter, J., 2001. The world's simplest grammars are Creole grammars. Linguist. Typol. 5 (2/3), 125–166.
Miestamo, M., Sinnemäki, K., Karlsson, F., 2008. Language Complexity: Typology, Contact, Change, vol. 94. John Benjamins Publishing.
Miller, G.A., Newman, E.B., Friedman, E.A., 1958. Length-frequency statistics for written English. Inf. Control 1 (4), 370–389.
Mostowski, M., 1998. Computational semantics for monadic quantifiers. J. Appl. Non-Classical Log. 8, 107–121.
Moxey, L., Sanford, A., 1993. Communicating Quantities. A Psychological Perspective. Lawrence Erlbaum Associates Publishers.
Peters, S., Westerståhl, D., 2006. Quantifiers in Language and Logic. Clarendon Press, Oxford.
Piantadosi, S.T., 2011. Learning and the Language of Thought (PhD thesis). Massachusetts Institute of Technology, Cambridge, Massachusetts.
Piantadosi, S.T., 2015. Zipf's word frequency law in natural language: a critical review and future directions. Psychonomic Bull. Rev. 21 (5), 1112–1130.
Ristad, E.S., 1993. The Language Complexity Game. The MIT Press.
Sampson, G., Gil, D., Trudgill, P., 2009. Language Complexity as an Evolving Variable, vol. 13. Oxford University Press.
Szymanik, J., 2010. Computational complexity of polyadic lifts of generalized quantifiers in natural language. Linguist. Philos. 33 (3), 215–250.

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

Szymanik, J., Zajenkowski, M., 2010. Comprehension of simple quantifiers. Empirical evaluation of a computational model. Cogn. Sci. Multidisciplinary J. 34 (3), 521–532.

Szymanik, J., Zajenkowski, M., 2013. Monotonicity has only a relative effect on the complexity of quantifier verification. In: Aloni, M., Franke, M., Roelofsen, F. (Eds.), Proceedings of the 19th Amsterdam Colloquium, pp. 219–225.

Thorne, C., 2012. Studying the distribution of fragments of English using deep semantic annotation. In: Proceedings of the ISA8 Workshop.