Using complexity to study linguistic expressiveness. A Case study of quantifiers in English and German.

Jakub Szymanik[1] and Camilo Thorne[2]

[1]Institute for Logic, Language and Computation, University of Amsterdam
[2]Data and Web Science Group, University of Mannheim

November 20, 2015

Abstract

We study semantic complexity of quantifiers, and their distribution in large-scale English and German corpora. The semantic complexity of a quantifier can be defined as the amount of computational resources necessary to decide whether the quantifier sentence is true in a finite situation or state of affairs. As it is known that the cognitive abilities (e.g., working memory) of speakers are limited, one would expect speakers to be biased towards quantifiers that are easy to compute (e.g. can be computed involving little to no working memory). We show that, as predicted by the theory, corpora distributions are significantly skewed towards the quantifiers of lower complexity. We also show that such correlation can be described by a power law.

1 Introduction

Linguists and philosophers have been searching for various ways to estimate complexity and expressivity 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., Mostowski and Szymanik, 2012)? What is the 'natural class of concepts' expressible in a given language and how to delimit it (see, e.g., Barwise and Cooper, 1981)? Are there differences between various languages with respect to semantic complexity (see, e.g., Everett, 2005)? Or more from a 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 also be asked from a cognitive angle: are some natural language concepts harder to process for humans than others (see Section 2.3)?

In order to contribute to the above outlined debate we focus on one aspect of natural language: its ability to express (often vague and relative) quantities by using a wide repertoire of quantifier expressions, like 'most', 'at least five', or 'all' (see, e.g., Keenan and Paperno, 2012). In the next sections we will focus on the semantic complexity of number concepts. This measure will deal with 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. Such idealized assumption (like many idealized assumptions in the sciences, e.g., point-masses in Newtonian physics) is both necessary (in that it simplifies analysis of a complex world and makes it independent from particular linguistic theories) and convenient (as it results in characterizations of phenomena that balance description simplicity with empirical adequacy).

In the last section we present linguistic experiments showing that semantic complexity can be used to predict strikingly similar distributions of quantifiers in both English and German textual data, i.e., in both corpora we find power law distributions with respect to the semantic complexity and frequency. Indeed, one of the linguistic reasons to expect power laws in natural language data is *the principle of least effort in communication*: speakers tend to minimize the communication effort by generating 'simple' messages. We take this result as a proof of concept, i.e., we claim that abstract semantic complexity measures (as the one considered in this paper) may enrich the methodological toolbox of the language complexity debate.

2 Semantic Complexity of Number Expressions

2.1 Quantifiers

What are the numerical expressions (quantifiers) 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./Über 7 Studenten sind klug.

(2) Fewer than eight students received good marks./Unter ein Halb der Studenten haben gute Bewertungen bekommen.

(3) More than half of the students danced nude on the table./Über ein Halb der Studenten haben genäckt getanzt.

(4) Fewer than half of the students saw a ghost./Unter ein Halb der Studenten haben einen Geist gesehen.

The formal semantics of natural language describes the meanings of these sentences. Sentences (1)–(4) share roughly the same linguistic form $\mathsf{Q}\ A\ B$, where Q is a quantifier (determiner), A is a predicate denoting the set of students, and B is another 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 ($\text{card}(A \cap B) > 7$). Analogously, the statement 'Fewer than eight A are B' is true if and only if $\text{card}(A \cap B) < 8$. 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. $\text{card}(A \cap B) > \text{card}(A - B)$) and then we can also formalize the meaning of sentence 'Fewer than half of the A are B' as $\text{card}(A \cap B) < \text{card}(A - B)$.[1]

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

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. Let us explain what we mean with the model below:

Imagine that you have a picture presenting colorful dots and consider the following sentence[2]:

(5) Every dot is red./Alle die Punkte sind rot.

[1] 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.

[2] As we will be considering English and German data, we provide examples in both languages.

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 Figure 1, which simply checks whether all elements are red.

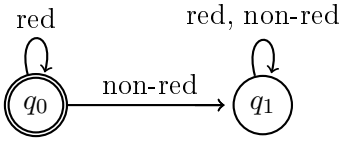


Figure 1: Finite automaton for the verification of sentence (5). It inspects the picture dot by dot starting in the accepting state (double circled), q_o. As long as it does not find a non-red dot it stays in the accepting state. If it finds such a dot, then it already 'knows' that the sentence is false and moves to the rejecting state, 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 q_0, however, we leave the loop on q_1 following the convention of completely defining the transition function.

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

(6) More than three dots are red./Über 3 Punkte sind rot.

(7) Fewer than four dots are red./Unter 4 Punkte sind rot.

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 finding four red elements, then statement (7) is true. We can easily compute the task using the following finite automata from Figures 2 and 3.[3]

[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,

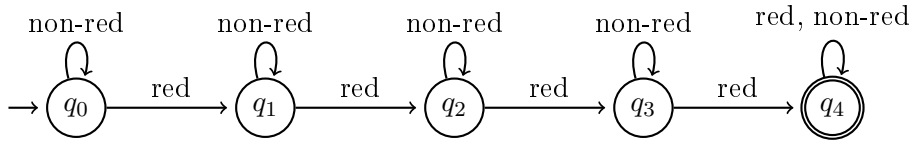


Figure 2: This finite automaton decides whether more than three dots are red. The automaton needs five states. It starts in the rejecting state, q_0, and eventually, if the condition is satisfied, moves to the double-circled accepting state, q_4. Furthermore, notice that to recognize 'more than seven',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

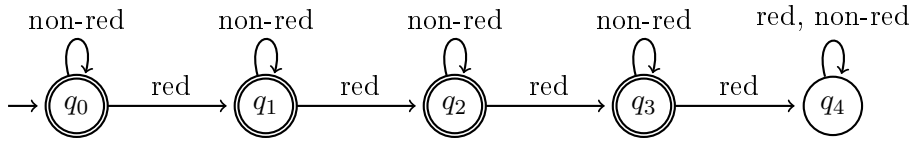


Figure 3: This finite automaton recognizes whether fewer than four dots are red. The automaton needs five states. It starts in the accepting state, q_0, and eventually, if the condition is not satisfied, moves to the rejecting state, q_4. Furthermore, notice that to recognize 'fewer than eight', we would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

These finite automata are very simple and they have only 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:

---

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).

(8) More than half of the dots are red./Über ein Halb der Punkte sind rot.

(9) Fewer than half of the dots are red./Unter ein Halb der Punkte sind rot.

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 develop 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 cannot 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 this 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 automaton 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.

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), see Table 1. In section 4 we show experimentally that this distinction correlates with linguistic data. But before we move to our experimental data let us briefly mention experimental cognitive evidence corroborating the significance of this distinction.

2.3 Quantifier Processing[4]

The above model of semantic complexity was suggested by Szymanik (2007) as a psychological model for some sentence-picture verification experiments in which subjects were asked to give precise judgments. It has been shown

[4]While reading this Section one may think about similar literature in Artificial Grammar Learning trying to assess the role of grammatical complexity in language inference (see, e.g., Schiff and Katan, 2014)

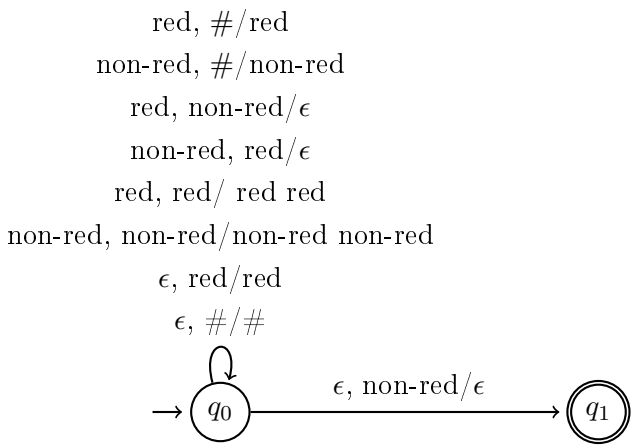


Figure 4: This push-down automaton recognizes whether fewer than half of dots are red. The automaton needs two states and the stack. It starts in the accepting state, q_0 with an empty stack marked by #. If it finds a red dot it pushes it on top of the stack and stays in q_0, if it finds a non-red dot 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=ϵ) 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.

that the computational distinction between quantifiers recognized by finite-automata and push-down automata is psychologically relevant, i.e., the more complex the automaton, the longer the reaction time and working memory involvement of subjects asked to solve the verification task: Szymanik and Zajenkowski (2010a) have shown that sentences with the Aristotelian quantifiers 'some' and 'every', corresponding to two-state finite automata, were solved in the least amount of time, while the proportional quantifiers 'more than half' and 'less than half' triggered the longest reaction times. When it comes to the numerical quantifiers 'more than k' and 'fewer than k', corre-

Table 1: Quantifiers and their semantic complexity. Note: we assume 'few' to be the dual of 'most', see also footnote 1.

class	examples	quantifier	complexity
Aristotelian	'every', 'some',	all, some	2-state acyclic FA
counting	'more than k', 'exactly 5'	>k, <k, k	k+2-state FA
proportional	'most', 'less than half' '10%', 'two-thirds' 'less than 3/5'	<p/k, p/k, >k/100, few <k/100, k/100 most, >p/k,	PDA

sponding to finite automata with $k + 2$ states, the corresponding latencies were positively correlated with the number k. Szymanik and Zajenkowski (2010b, 2011) have explored this complexity hierarchy in concurrent verification experiments, and have shown that during the verification, the subjects' working memory is qualitatively more engaged while verifying proportional quantifiers than while verifying numerical and Aristotelian quantifiers. Actually, McMillan et al. (2005), in an fMRI study, have shown that during verification, all sentences activate the right inferior parietal cortex associated with numerosity, but proportional quantifiers activate also the prefrontal cortex, which is associated with executive resources, such as working memory. These findings were later strengthened by the evidence on quantifier comprehension in patients with focal neurodegenerative disease (McMillan et al., 2006). Moreover, recently Zajenkowski et al. (2011) have compared the verification of natural language quantifier sentences in a group of patients with schizophrenia and a healthy control group. In both groups, the difficulty of the quantifiers was consistent with the computational predictions, even if patients with schizophrenia took more time to solve the problems. However, they were significantly less accurate only with proportional quantifiers, such as 'more than half'. Finally, Zajenkowski and Szymanik (2013) have explored the relationship between intelligence, working memory, executive functions and complexity of quantifiers to find out that the automata model

nicely predicts the correlations between those various measures of cognitive load. All this evidence speaks in favor of the thesis that the model can capture some cognitive aspect of the semantics for generalized quantifiers. However, these studies have exclusively focused on the complexity of verification procedures for various quantifier sentences, hence, the question arises as to whether the distinction between regular and context-free quantifiers is also reflected in very large English and German corpora, large enough to be considered representative for either language. In the next sections we show that this appears to be the case.

3 Power Laws

We believe that semantic complexity has an observable impact on quantifier use by speakers, which can be harnessed and quantified using Zipfean relations or power laws. Power laws in natural language data were first discovered by the American linguist and statistician George K. Zipf in the early 20th century. Power laws are non-normal skewed distributions where, intuitively, the topmost 20% outcomes of an ordinal variable concentrate around 80% of the probability mass or frequency. They are widespread in natural language data (cf. Baroni, 2009).

Zipf further hypothesized that power laws and, in general, biased distributions arise in natural language data due to the so-called *principle of least effort* in human communication: Speakers seek to minimize their effort to generate a message by using few, short, ambiguous words and short sentences. While hearers seek to minimize their effort to understand a message by requiring the opposite. This typically gives rise to textual datasets or *corpora* in which, while encompassing large vocabularies, a small subset of words is used very frequently.

More recent work (cf. Newman, 2005) has shown that Zipf's original equations can be modified to cover a larger spectrum of natural language phenomena, suggesting that Zipf's principle may apply not only to surface features such as length or vocabulary size, but also to deep features such as computational complexity. In what follows we will endeavor to show that low complexity quantifiers occur more likely than high complexity quantifiers. Furthermore, that the following *power law* or Zipfean relation between *quantifier frequency* $fr(\mathsf{Q})$ and *quantifier rank* $rk(\mathsf{Q})$ described by the equa-

Table 2: Corpora used in this study.

corpus	sentences	tokens
Sdewac (Ger)	$\sim$ 45 million	$\sim$ 800 million
WaCkY (Eng)	$\sim$ 43 million	$\sim$ 800 million

tion

$$fr(\mathsf{Q}) = a/rk(\mathsf{Q})^b \tag{PL}$$

can be observed in very large corpora. For the purposes of this paper we assume $rk(\mathsf{Q})$ to be an ordered factor (see Table 1), with quantifiers ordered by their semantic complexity and expressiveness, whereas $fr(\mathsf{Q})$ refers to their *raw* frequency (absolute counts).

Power laws are inferred by estimating (statistically) their coefficients or parameters. Many techniques are possible to estimate their parameters. To approximate the parameters a and b in (PL) we relied in our experiments on the standard least squares linear regression technique (see (Newman, 2005)). This is because power laws are equivalent to linear models on the log-log scale:

$$fr(\mathsf{Q}) = a/rk(\mathsf{Q})^b$$
$$\text{iff}$$
$$\log(fr(\mathsf{Q})) = a - b \cdot \log(rk(\mathsf{Q})).$$

4 Experiment

In this section we outline our analysis of generalized quantifier frequency in corpora. We approximated our quantifiers' distribution by identifying their surface forms in two large corpora built from the English and German Wikipedias. In addition, we checked if such distribution, as discussed in Section 3 is skewed towards finite-automata quantifiers and can be described by a power law. Given that negation in general has no impact in semantic complexity (as understood in our paper), we disregarded negative quantifiers and all (natural language) polarity issues. Furthermore, rather than covering all the linguistic aspects of the quantifiers studied –a considerable challenge that goes beyond the scope of this paper–, we focused on their main surface forms and lexical variants.

4.1 Corpora

To obtain a representative sample, we considered two very large English and German corpora covering multiple domains and sentence types (declarative and interrogative). Specifically, we considered two corpora, built and curated by Baroni et al. (2009), the WaCkY (English) and Stuttgart Sdewack (German) corpora. Both corpora were built by postprocessing full dumps (from 2010) of Wikipedia. The authors removed all HTML markup and image files, and filtered out those webpages devoid of real textual content (e.g., tables displaying statistics), until balanced (relatively to subject matter or domain, vocabulary, sentence type and structure, etc.) corpora representative of English and Germen were achieved. See Table 2 for details on their size; for full details, please refer to Baroni et al. (2009).

The WaCkY corpus was segmented, tokenized and linguistically annotated using the TreeTagger statistical parser[5], that has an accuracy of over 90% for both languages, resulting in datasets that exhibit the format shown in Figure 5. For each sentence, the corpora provide the following information: *(i)* the list of its tokens (first column), *(ii)* the list of their corresponding lemmas or morphological stems (second column), *(iii)* the list of their corresponding part-of-speech (POS) tags (third column). The WaCkY corpus provides in addition: *(iv)* information regarding the position of the words in the sentence (fourth and fifth columns), and *(v)* the list of their corresponding typed (syntactic) dependencies (fith column). For our experiments, we took into consideration only *(i)–(iii)*, shared by both corpora. The POS tags used by TreeTagger (for both English and German) are derived from the well-known Penn Treebank list of POS tags.[6]

4.2 Patterns

We identify generalized quantifiers indirectly, via part-of-speech (POS) patterns (regular expressions) that approximate their surface forms. Each such pattern defines a quantifier *type*, modulo *lexical variants*. In what follows, we counted the number of times each type is instantiated within a sentence in the corpus, that is, its number of *tokens*.

Notice that to properly identify surface forms, POS tags are necessary,

[5] `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`

[6] `http://www.ling.upenn.edu/courses/Fall_2007/ling001/penn_treebank_pos.html`.

```
<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>
```

Figure 5: Sample tokenized, POS-annotated sentence from the WaCkY corpus.

given the peculiarities of our datasets. For instance, the Aristotelian quantifier (type) *all* is usually expressed in the Baroni corpora by the determiner (DT) 'every', but sometimes by the determiner 'the' followed by a plural noun (NNS) as in 'the men' (as short for 'all the men') Furthermore, notice that lexical variants are key to identifying quantifiers, since, in general, many different surface forms may be used to denote them. Thus, *some* is not only expressed or denoted by the POS-annotated surface form 'some/DT', but also by pronouns such as 'somebody', viz., by surface forms such as 'somebody/PN'.

Table 3 provides an overview of the patterns considered for the experiment described in this paper. Every cluster of patterns gave rise to regular expressions that were run over the corpora. Their rationale was to capture quantifier lexical variants In what follows we give two examples of what we mean by lexical variants:

(1) To identify the Aristotelian quantifier 'all' in English, we considered its lexical variants 'all', 'everybody', 'everything', 'every', 'each', 'everyone' and 'the N', where N stands for a plural noun.

(2) To identify the Aristotelian quantifier "all" in German, we considered

its lexical variants 'einige', 'jemand', 'etwas', 'irgendetwas', 'ein', 'es gibt', 'manche' and 'viel'.

Notice that we lowercased all the input sentences and words and focused on lemmas whenever possible to avoid unnecessarily multiplying patterns due to inflection (particularly in German).

4.3 Model Validation

To validate our models, we computed the R^2 coefficient, that measures how well a set of observations fits an inferred power law equation, and ranges from 0 (no fit) to 1 (perfect fit). If the coefficient is higher than 0.9, then we can say with high confidence that a distribution follows a power law (cf. Newman, 2005).

Secondly, we tested if the distributions observerd (and their bias) were random phenomenona or described some real pattern inherent to our datasets. To this end we run a χ^2 test (at $p = 0.01$ significance) w.r.t. the uniform distribution as our null hypothesis (cf. Gries, 2010).

Finally, we measured the skewness of the distribution (cf. Gries, 2010). Skewness is a statistical measure that quantifies how much the distribution is symmetrical (which would be the case if it were Gaussian). A positive value indicates a bias in (probability) density towards the y-axis, viz., the first/highest ranked p outcomes of the (random) ordinal variable V whose distribution we are analyzing. A negative value, the converse bias. The higher the value, the higher the bias. Finally, a value close to 0 indicates a normal distribution.

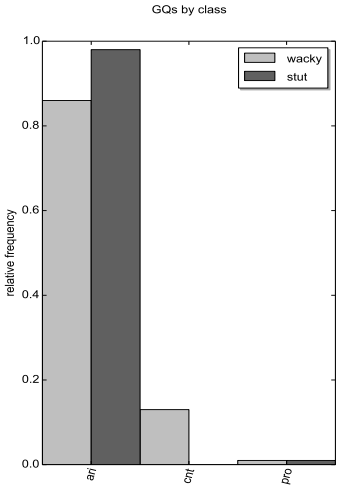
4.4 Results and Interpretation

The distributions observed are summarized by Figures 6 and 7. The reader will find on the left of Figure 7 the relative average and cumulative frequency plots for the quantifiers considered, and to the right the plots of the log-log regressions. They also provide the contingency tables from which the plots were generated, and the results of the statistical tests. Finally, observe that Figure 7, top right, spells out the power law/Zipfean relations inferred in addition to the model validation results.

As expected by the theory and our assumptions, Aristotelian quantifiers are more frequent than counting quantifiers, and counting quantifiers than proportional quantifiers. Moreover, the trend appears to be cross-linguistic

Table 3: Quantifiers studied in this paper and patterns considered. Notice the use of lemmas for German.

all	some	>k	<k
every/dt, all/dt, the/dt .*/nns, everything/nn, everyone/nn everybody/nn, each/dt, no/dt,	someone/nn, somebody/nn, anybody/nn, something/nn, some/dt a/dt, many/dt, many/jj */nns, there/ex	at/in least/jjs .*/cd more/jjr than/in .*/cd more/jjr than/in .*/at .*/cd	at/in most/jjs .*/cd less/jjr than/in .*/cd fewer/jjr than/in .*/at .*/cd less/jjr than/in .*/at .*/cd fewer/jjr than/in .*/at .*/cd
piat/alle, pis/alle, piat/kein piat/jed	pis/jemand, pis/etwas, piat/etwas, art/ein pper/es vvfnn/gibt, pis/manch, piat/manch, piat/viel ne/irgendetwas	adv/mindestens card/@card@ piat/mehr kokom/als card/@card@	adv/h\p{L}chstens card/@card@ piat/weniger kokom/als card/@card@
k	most	>p/k	
.*/cd .*/nns exactly/rb .*/cd	most/jjs most/dt more/jjr than/in half/nn	more/ap than/in half/abn more/ap than/in .*/cd of/in	
card/@card@ nn/.*	adv/fast piat/jed piat/mehr kokom/als adjd/halb appr/\p{L}ber adjd/halb	piat/mehr kokom/als adjd/halb appr/\p{L}ber adjd/halb piat/mehr kokom/als card/@card@ appr/von appr/\{L}ber card/@card@ appr/von	
<p/k	<k/100	p/k	>k/100
less/jjr than/in half/nn fewer/jjr than/in half/nn less/jjr than/in .*/cd of/in fewer/jjr than/in .*/cd of/in	less/jjr than/in .*/cd percent/nn less/jjr than/in %/cd "	half/dt, half/pdt, half/nn of/in .*/nns of/in, .*/nn of/in	more/jjr than/in .*/cd percent/nn more/jjr than/in %/cd
piat/weniger kokom/als adjd/halb piat/weniger kokom/als card/@card@ appr/von appr/unter card/@card@ appr/von appr/unter adjd/halb	appr/unter card/@card@ nn/% piat/weniger kokom/als card/@card@ nn/%	adja/halb adja/halb appr/von card/@card@ appr/von	appr/uber card/@card@ nn/% piat/mehr kokom/als card/@card@ nn/%
few	k/100		
few/jj, few/dt less/jj than/in half/nn fewer/jj than/in half/nn	./cd percent/nn, %/cd		
piat/wenig piat/wenig kokom/als adjd/halb appr/unter adjd/halb	nn/%		



test	value
skewness (means)	0.7
χ^2-test ($p = 0.01$ sig.)	$p = 0.0$

	pro	cnt	ari
WaCkY	888709	8362986	57355333
Sdewac	595543	237718	48138864
total	1484252	8600704	105494197

Figure 6: Left: Quantifier distribution by quantifier class. Right: Raw frequencies per corpus.

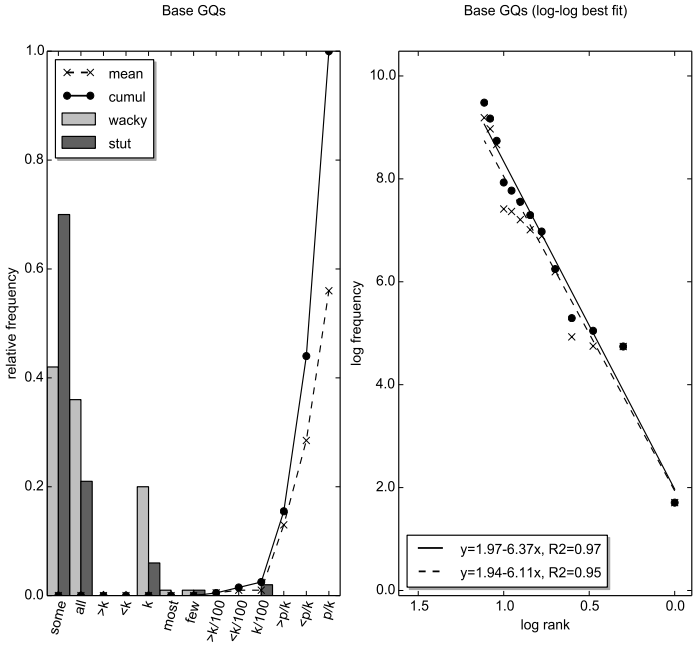
(since shared by both corpora). Figure 6, right, shows that this bias is statistically strongly significant: their distribution significantly differs from uniform or random distributions (the null hypothesis rejected by the test), since $p < 0.01$. Their distribution shows also a high measure of skewness.

Furthermore, we can infer power laws wherein Aristotelian quantifiers represent $> 80\%$ of the (mean) frequency mass. See Figure 7. Indeed, a high goodness-of-fit coefficient was obtained: $R^2 = 0.94$. The distribution is again statistically significant and exhibits an even greater measure of skewness.

5 Conclusions

Our results, together with Thorne (2012), show that abstract computational complexity measures allow quantifying the complexity of natural language and suggest that their distribution in large textual datasets follows a power law or Zipfian relation relatively to their semantic (data) complexity. The usefulness of computational approaches to assess the intricate complexity of linguistic expressions gathers additional support from experimental studies in psycholinguistics.

The results also contribute to the discussion of semantic universals for natural language quantifiers (see Barwise and Cooper, 1981; Peters and West-



model/test	value
power law (cumul.)	$fr(\mathsf{Q}) = 1.97/rk(\mathsf{Q})^{6.37}$
power law (means)	$fr(\mathsf{Q}) = 1.94/rk(\mathsf{Q})^{6.11}$
R^2 coeff. (cumul.)	$R^2 = 0.97$
R^2 coeff. (means)	$R^2 = 0.95$
skewness (means)	1.95
χ^2-test ($p = 0.01$ sig.)	$p = 0.0$

	>k	<k	most	>p/k	<p/k	p/k	>k/100	<k/100	few	k/100	k	all	some
WaCkY	198417	31337	312815	0	1086	69782	1698	1044	248478	3066	8133232	14646544	17013856
Sdewac	6847	0	9320	0	16	84298	0	80	270958	461742	1135029	4089797	13840686
total	205264	31337	322135	0	1102	154080	1698	1124	519436	464808	9268261	18736341	30854542

Figure 7: Top left: Quantifier distribution and power law regression. Top right: Summary of statistical tests. Bottom: Raw frequencies per corpus.

erståhl, 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 but also on the computational complexity of underlying semantic concepts. Simply speaking, some quantifiers may not be realized in natural language (or be used very rarely) due to their semantic complexity.[7]

As we mentioned in the introduction our goal was to give a proof of concept as for the applicability of abstract computational complexity measures in quantifying semantic complexity. As the next step we would like to use semantic complexity in the discussion of the *equivalent complexity thesis*: all natural languages are equally complex (have equal descriptive power) (see, e.g., Miestamo et al., 2008). The debate whether language complexity is a universal constant surely has great general importance and demands careful methodological scrutiny. The notion of semantic complexity explored here (or some of its variants) could be used to enrich the methodological toolbox used in this debate. For instance, as a first step, we could compare some Western languages with some Creole languages with respect to our complexity distinctions, i.e., check whether all languages realize equally complex (e.g., context-free) semantic constructions, like proportional quantifiers, and whether they have similar distributions (realize equally complex expressions equally often). In that way we could contribute to the debate whether creole languages are simpler than other languages.

References

Baroni, M. (2009). Distributions in text. In *Corpus linguistics: An International Handbook*, volume 2, pages 803–821. Mouton de Gruyter.

Baroni, M., Bernardini, S., Ferraresi, A., and Zanchetta, E. (2009). The WaCky Wide Web: A collection of very large linguistically processed web-crawled corpora. *Language Resources and Evaluation*, 43(3):209–226.

Barwise, J. and Cooper, R. (1981). Generalized quantifiers and natural language. *Linguistics and Philosophy*, 4:159–219.

[7] For an example see the discussion of collective quantifiers in Kontinen and Szymanik (2008) or reciprocal expressions in Szymanik (2010). Of course, there are other factors at play than only computational complexity. For instance, as pointed out by Hedde Zeijlstra one of the most famous quantifier that is never attested is 'nall' ('not all') which is actually very simplex in terms of complexity.

van Benthem, J. (1986). *Essays in logical semantics*. Reidel.

Everett, D. (2005). Cultural constraints on grammar and cognition in Pirahã. *Current Anthropology*, 46(4):621–646.

Gries, S. T. (2010). Useful statistics for corpus linguistics. In Sánchez, A. and Almela, M., editors, *A mosaic of corpus linguistics: selected approaches*, pages 269–291. Peter Lang.

Hopcroft, J. E., Motwani, R., and Ullman, J. D. (2000). *Introduction to Automata Theory, Languages, and Computation*. Addison Wesley, 2nd edition.

Juola, P. (1998). Measuring linguistic complexity: The morphological tier. *Journal of Quantitative Linguistics*, 5(3):206–213.

Keenan, E. L. and Paperno, D. (2012). *Handbook of quantifiers in natural language*, volume 90. Springer.

Kontinen, J. and Szymanik, J. (2008). A remark on collective quantification. *Journal of Logic, Language and Information*, 17(2):131–140.

McMillan, C. T., Clark, R., Moore, P., Devita, C., and Grossman, M. (2005). Neural basis for generalized quantifier comprehension. *Neuropsychologia*, 43:1729–1737.

McMillan, C. T., Clark, R., Moore, P., and Grossman, M. (2006). Quantifiers comprehension in corticobasal degeneration. *Brain and Cognition*, 65:250–260.

McWhorter, J. (2001). The world's simplest grammars are creole grammars. *Linguistic Typology*, 5(2/3):125–166.

Miestamo, M., Sinnemäki, K., and Karlsson, F., editors (2008). *Language Complexity: Typology, contact, change*. Studies in Language Companion Series. John Benjamins Publishing Company.

Mostowski, M. (1998). Computational semantics for monadic quantifiers. *Journal of Applied Non-Classical Logics*, 8:107–121.

Mostowski, M. and Szymanik, J. (2012). Semantic bounds for everyday language. *Semiotica*, 188(1-4):363–372.

Newman, M. E. J. (2005). Power laws, Pareto distributions and Zipf's law. *Contemporary Physics*, 46(5):323–351.

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

Ristad, E. S. (1993). *The Language Complexity Game*. The MIT Press.

Sampson, G., Gil, D., and Trudgill, P. (2009). *Language complexity as an evolving variable*, volume 13. Oxford University Press.

Schiff, R. and Katan, P. (2014). Does complexity matter? meta-analysis of learner performance in artificial grammar tasks. *Frontiers in Psychology*, 5(1084).

Szymanik, J. (2007). A comment on a neuroimaging study of natural language quantifier comprehension. *Neuropsychologia*, 45(9):2158–2160.

Szymanik, J. (2010). Computational complexity of polyadic lifts of generalized quantifiers in natural language. *Linguistics and Philosophy*, 33(3):215–250.

Szymanik, J. and Zajenkowski, M. (2010a). Comprehension of simple quantifiers. Empirical evaluation of a computational model. *Cognitive Science: A Multidisciplinary Journal*, 34(3):521–532.

Szymanik, J. and Zajenkowski, M. (2010b). Quantifiers and working memory. In Aloni, M. and Schulz, K., editors, *Amsterdam Colloquium 2009, Lecture Notes In Artificial Intelligence 6042*, pages 456–464. Springer.

Szymanik, J. and Zajenkowski, M. (2011). Contribution of working memory in parity and proportional judgments. *Belgian Journal of Linguistics*, 25(1):176–194.

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

Zajenkowski, M., Styła, R., and Szymanik, J. (2011). A computational approach to quantifiers as an explanation for some language impairments in schizophrenia. *Journal of Communication Disorders*, 44(6):595 – 600.

Zajenkowski, M. and Szymanik, J. (2013). Most intelligent people are accurate and some fast people are intelligent: Intelligence, working memory, and semantic processing of quantifiers from a computational perspective. *Intelligence*, 41(5):456 – 466.